

Scalable Nonlinear Compact Schemes

Mathematics and Computer Science Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Availability of This Report:

This report is available, at no cost, at <http://www.osti.gov/bridge>. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

Scalable Nonlinear Compact Schemes

prepared by:

Debojyoti Ghosh¹, Emil M. Constantinescu^{1,2}, and Jed Brown^{1,3}

¹Mathematics & Computer Science Division, Argonne National Laboratory

²Computation Institute, University of Chicago

³Department of Computer Science, University of Colorado Boulder

April 2014

Contents

- Executive Summary ii
- 1 Introduction 1
- 2 Compact Finite-Difference Schemes 4
 - 2.1 WENO Schemes 5
 - 2.2 CRWENO Schemes 6
 - 2.3 Boundary Treatment 7
 - 2.4 Numerical Properties 8
- 3 Parallel Implementation 9
 - 3.1 Tridiagonal Solver 10
 - 3.2 Performance Analysis 15
- 4 Results 28
 - 4.1 Isentropic Vortex Convection 29
 - 4.2 Isotropic Turbulence Decay 31
- 5 Conclusions 34
- References 36

Executive Summary

Solutions to hyperbolic conservation laws are often characterized by a large range of length scales as well as discontinuities. Standard nonlinear finite-difference schemes, such as the WENO schemes, yield non-oscillatory solutions but lack the spectral resolution required to model the relevant length scales. Linear compact schemes have a high spectral resolution; however, they suffer from spurious oscillations across discontinuities and sharp gradients. Weighted nonlinear compact schemes, such as the CRWENO scheme and the hybrid compact-WENO schemes, combine the non-oscillatory nature of the WENO schemes with the high spectral resolution of the compact schemes and are thus ideal for solutions with multiple length scales and discontinuities. One example of an application area is compressible, turbulent flows. The CRWENO scheme and the hybrid compact-WENO schemes have a nonlinear, solution-dependent left-hand side and therefore require the solution of banded systems of equations at each time-integration step or stage. Application of these schemes to multiprocessor simulations requires an efficient, scalable algorithm for the solution to the banded systems. Past efforts at implementing nonlinear compact schemes for parallel simulations suffer from one or more of the following drawbacks: parallelization-induced approximations and errors, complicated and inefficient scheduling of communication and computation, significant increase in the mathematical complexity of the banded systems solver, and high communication overhead. Therefore, these algorithms do not scale well for massively parallel simulations and are inefficient compared with the corresponding standard finite-difference schemes.

In this work, we focus on compact schemes resulting in tridiagonal systems of equations, specifically the fifth-order CRWENO scheme. We propose a scalable implementation of the nonlinear compact schemes by implementing a parallel tridiagonal solver based on the partitioning/substructuring approach. We use an iterative solver for the reduced system of equations; however, we solve this system to machine zero accuracy to ensure that no parallelization errors are introduced. It is possible to achieve machine-zero convergence with few iterations because of the diagonal dominance of the system. The number of iterations is specified a priori instead of a norm-based exit criterion, and collective communications are avoided. The overall algorithm thus involves only point-to-point communication between neighboring processors. Our implementation of the tridiagonal solver differs from and avoids the drawbacks of past efforts in the following ways: it introduces no parallelization-related approximations (multiprocessor solutions are exactly identical to uniprocessor ones), it involves minimal communication, the mathematical complexity is similar to that of the Thomas algorithm on a single processor, and it does not require any communication and computation scheduling.

We analyze the performance of the fifth-order CRWENO scheme with this parallel tridiagonal solver and compare its computational efficiency on multiple processors with that of the WENO scheme. The Euler equations governing inviscid, compressible flows are solved in one, two, and three spatial dimensions and we consider two types of test problems—the advection of a smooth sinusoidal density waves and the advection of density fluctuations comprising all grid-supported wavelengths. We demonstrate that in one and two spatial dimensions, there exists a critical subdomain size above which the compact schemes are more efficient (for smaller subdomains, i.e., larger number of processors, the WENO scheme is more efficient). In three spatial dimensions, however, the compact schemes are more efficient for the entire range of parallelization. We consider two benchmark physically-relevant flow problems—long-term convection of an isentropic vortex and the decay of isotropic turbulent—and evaluate the strong and weak scalability of our algorithm on up to $\sim 500,000$ processors. We show that the fifth-order CRWENO scheme retains its higher computational efficiency compared with that of the WENO scheme, for the smallest subdomain sizes possible.

1 Introduction

Hyperbolic conservation laws govern the dynamics of several physical applications that involve convection or wave propagation [44]. The governing equations express the conservation of a quantity by equating the time derivative of that quantity inside a control volume to its spatial flux through the boundaries. Examples of such systems include compressible fluid dynamics, ideal magnetohydrodynamics, and electromagnetics. Linear hyperbolic partial differential equations (PDEs) admit discontinuities in their solution if the initial or boundary conditions have discontinuities, while for nonlinear hyperbolic PDEs, discontinuities may form from smooth initial and boundary conditions. Numerical methods for the solution of such equations need to yield non-oscillatory solutions across discontinuities while maintaining high-order accuracy for smooth solutions. Several methods have been proposed in the literature [39, 44], including the first-order schemes of Godunov [30] and Roe [54] and second-order schemes such as the MUSCL scheme [67], Harten’s TVD scheme [33], and the piecewise parabolic method [15].

The essentially non-oscillatory (ENO) schemes [34] use an adaptive stenciling procedure to achieve high-order accuracy for smooth solutions and avoid oscillations across discontinuities. The interpolation stencil with the smoothest data (as measured by divided differences) is selected from among the candidate stencils to compute the final interface flux, thus avoiding stencils that contain discontinuities. The implementation of the ENO schemes was further improved [60, 61] for conservative finite-difference and finite-volume schemes and applied to several practical applications. The weighted ENO (WENO) schemes were introduced [45] as an improvement where the final interface flux is calculated by using a weighted combination of the candidate stencils. The weights are functions of the solution smoothness and approach zero for stencils containing discontinuities. Away from discontinuities, the weights approach their optimal values, and a higher-order accurate interpolation of the interface flux is obtained. The implementation of the WENO schemes were improved [37], and very high-order accurate WENO schemes were constructed [8]. The original formulation of the solution-dependent weights was observed to be excessively dissipative and result in suboptimal convergence for some classes of smooth problems, and alternative nonlinear weights were proposed [10, 12, 35, 72, 73]. These formulations led to significant improvements in the behavior of the WENO schemes. The WENO schemes have been successfully applied to several applications [59].

Solutions to hyperbolic conservation laws are often characterized by a large range of length scales. One example is compressible turbulent flow where the length scales range from the characteristic scale of the flow to that of the turbulent flow structures. Numerical algorithms for such applications need a high spectral resolution to accurately model all the relevant scales; and one of the primary drawbacks of the WENO schemes (and other standard finite-difference schemes) is poor spectral resolution, even for very high orders of accuracy. While spectral methods [11, 31] capture a given range of scales exactly, their applications are restricted to simple domains with periodic boundaries. Compact finite-difference (FD) schemes were introduced [42] that had significantly higher spectral resolution compared with that of noncompact schemes with the same order of convergence. These schemes use an implicit formulation for evaluating the interface flux or the cell-centered flux derivative to achieve higher orders of convergence with smaller stencil sizes (hence the term “compact”) and higher spectral resolution for a given order of convergence. The schemes have been successfully applied to incompressible [20, 71] and compressible [21, 43] fluid dynamics, aero-acoustics [21, 56], and electromagnetics [57]. The high spectral resolution has led to their application to direct numerical simulation (DNS) [40] and large-eddy simulation (LES) [50] of turbulent flows. Originally formulated for a uniform grid, they have been extended and applied to nonuniform grids [24, 62, 68]. Linear compact schemes result in oscillations across discontinuities and a non-linearly stable, total variation bounded (TVB) compact scheme was proposed [14] for shock calculations and further improved [74].

Several attempts have been made to construct a high-resolution, non-oscillatory scheme by applying the solution-adaptive algorithm of the WENO scheme to compact schemes. One approach to construct such schemes, the weighted compact nonlinear schemes (WCNS), was proposed [17, 18] based on a two-stage nonconservative computation of the flux derivative on a staggered mesh. The first stage involves the non-oscillatory reconstruction of the flux at the interface using an adaptive compact scheme [17] or the WENO scheme [18], and the second stage computes the cell-centered flux derivative from the interface fluxes using a

high-order central compact scheme. The use of an adaptive compact scheme in the first stage [17] required the solution to three tridiagonal systems of equations at each time-integration step/stage. This drawback resulted in the use of the noncompact ENO or WENO schemes for the first stage in subsequent implementations of this approach [18, 70, 75]. Thus, the overall scheme has a linear left-hand side and a solution-dependent right-hand side, and the LU decomposition of the left-hand side can be carried out as a precomputation step. Although this approach is computationally inexpensive, the use of noncompact schemes in the interface flux calculations compromises the spectral resolution of the overall scheme. It is observed [18, Figure 2] that the spectral resolution of these schemes is only marginally higher than that of the WENO schemes.

Another approach for implementing a weighted, non-oscillatory compact scheme is the hybrid compact-ENO/WENO scheme. A hybrid compact-ENO scheme [4] was introduced in which the ENO scheme is used at grid points at and near a discontinuity while an upwind compact scheme is used at all other grid points. A first-order discontinuity detector is used to switch between the compact scheme and the ENO scheme. Two upwind compact schemes were introduced requiring the solution of pentadiagonal systems of equations. This algorithm was applied to the DNS of turbulent flow over a compression ramp [3]. This approach was improved by introducing the hybrid compact-WENO scheme [51] and formulating the compact scheme as a conservative finite-difference scheme to improve the coupling with the shock-capturing WENO scheme. The computational cost was reduced by using a compact scheme that requires the solution to a tridiagonal system of equations. A characteristic-based hybrid compact-WENO scheme was proposed [53] in which the reconstruction is carried out on the characteristic variables; however, this requires the solution of a block tridiagonal system of equations and is computationally more expensive. In addition, a smooth switching parameter is used to switch between the compact scheme and the WENO scheme. One of the primary drawbacks of this approach is that it uses a non-compact scheme at and near discontinuities, thus reducing the spectral resolution. This is critical for studies involving the interaction of small length-scale features with discontinuities, such as shock-turbulence interactions. In addition, for flows dominated by a large number of discontinuities and sharp gradients (e.g., shocklets and shear layers in fluid dynamics), this approach will result in applying the noncompact ENO or WENO scheme to the majority of the domain.

A third approach is the compact-reconstruction WENO (CRWENO) scheme [25, 27] that applies the WENO algorithm to compact interpolation schemes. Lower-order compact schemes are identified along with optimal coefficients such that their weighted sum is a higher-order compact scheme. Solution-dependent weights are computed by using the smoothness indicators of the WENO scheme such that the overall scheme is non-oscillatory across discontinuities and high-order accurate for smooth solutions. Unlike the hybrid compact-WENO schemes, the CRWENO schemes reduce to a lower-order compact scheme near the discontinuities, instead of a noncompact scheme, and thus retain a higher spectral resolution. Fifth-order CRWENO schemes have been constructed and successfully applied to a large range of problems, including the DNS of turbulent flows [26] as well as compressible aerodynamics flows [28]. Although the CRWENO scheme uses compact interpolation stencils, the calculation of the nonlinear weights is identical to that of the WENO scheme, and the overall scheme has the same stencil width as the WENO scheme.

While the WCNS schemes have a linear left-hand side that can be prefactored, the hybrid compact-WENO and the CRWENO schemes require the solution of a solution-dependent banded system of equations along each one-dimensional grid line. The hybrid compact-WENO scheme uses a linear compact scheme; however, the position of discontinuities may change as the solution evolves, and thus the grid points where the WENO scheme is applied changes. The final system of equations depends on the solution at that specific time. The CRWENO scheme results in a solution-dependent system of equations since the coefficients are a function of the local smoothness of the solution. The hybrid compact-WENO and the CRWENO schemes thus entail the added numerical cost of solving banded systems of equations along each grid line at each time-integration step or stage. It has been demonstrated [27] that the CRWENO scheme is more computationally efficient than the WENO scheme for single-processor simulations. Although the CRWENO scheme is costlier for the same grid size (because of the solution of the banded system of equations), it yields solutions with significantly lower errors that are comparable to those of the WENO scheme on a finer grid. The CRWENO scheme was thus less expensive when comparing solutions of the same accuracy and resolution. This result was demonstrated on smooth problems as well as problems with discontinuities for scalar conservation laws and systems of

equations with component-wise reconstruction. Although a quantitative analysis of the numerical cost of hybrid compact-WENO has not yet been presented in the literature, the conclusions regarding the CRWENO scheme are expected to hold true also for the hybrid schemes, since the mathematical complexity is similar.

Thus, a crucial issue in the scalable, parallel implementation of these nonlinear compact schemes (the hybrid compact-WENO and the CRWENO schemes) on distributed-memory platforms is the efficient parallel solution of banded systems of equations (in this study, we focus entirely on compact schemes that result in a tridiagonal system). Previous attempts at implementing parallel compact schemes have followed three approaches. One approach is the decoupling of the global system of equations into separate systems inside each subdomain. A parallel hybrid compact-WENO scheme was proposed [13] in which the noncompact WENO scheme is used to calculate the interface flux at the subdomain boundaries. This results in decoupled systems of equations inside each subdomain that can be solved in parallel. The communication requirements for this approach is identical to a corresponding noncompact scheme. A similar approach was proposed [38] in which a central compact scheme is used in the interior and biased compact schemes are used at the physical and subdomain boundaries to eliminate any coupling across the subdomain boundaries. The boundary schemes are derived to minimize the numerical instabilities arising from applying a downwind scheme at the inflow boundary and the excessive dissipation due to the upwind scheme at the outflow boundary. In order to yield non-oscillatory solutions, however, a compact filter is used that requires the solution of a global system of equations at each time-integration step and is solved in parallel using a predictor-corrector approach. These attempts were successful in obtaining high-resolution solutions with compact schemes on parallel platforms; however, one major drawback is that the decoupling across subdomains results in the numerical properties of the scheme being a function of the number of processors. As the number of processors increase for the same domain size, the spectral properties of the compact scheme gets compromised (more so in [13] since the noncompact WENO scheme is used at the subdomain boundaries), and numerical errors are introduced [38]. Thus, this approach is not well suited for massively parallel simulations with small subdomain sizes.

An alternative approach has been the parallel implementation of the tridiagonal solver. A parallel compact scheme with a pipelined Thomas algorithm (PTA) [52] was proposed in which the idle time of the processors during the forward and backward solves is used to carry out nonlocal data-independent computations (such as solving in the next spatial dimension) or local data-dependent Runge-Kutta step completion calculations. This algorithm requires a complicated static schedule to control the processor communications and computations. In addition, there is a trade-off between communication and computation efficiencies: either smaller messages are communicated, resulting in smaller idle times for the processors or larger messages are communicated but with longer idle times. A reduced parallel diagonally dominant (PDD) [64] algorithm was developed that solves a perturbed linear system. This approach introduces an error due to assumption of diagonal dominance that is bounded. In addition, several other efforts have been made to implement a parallel tridiagonal solver [19, 23, 36, 48, 49, 63, 69], although they may not have been applied specifically to compact finite-difference schemes. Some of these algorithms suffer from the drawback that they significantly increase the number of mathematical operations required to solve the system [36].

A data transposition method was proposed [16] in which the entire system of equations to be solved is collected on one processor and solved sequentially. In a multidimensional simulations, for dimensions along which the domain has been parallelized, “pencils” of data are transposed between processors. A similar approach has been used for pseudo-spectral schemes [29]. This approach suffers from a major drawback that it is very communication intensive; as noted in both these studies, over 60% of the total execution time is spent in the data transposition operations. This makes the efficiency and speedup of the algorithms dependent on the communication networks of the parallel platform.

The various drawbacks of the current approaches have limited the application of compact finite-difference schemes in the context of massively parallel simulations in application areas such as turbulent flows. Recent work on leadership-class computing platforms, such as [9], apply standard, noncompact finite-difference schemes. In this work, we seek to address this deficiency. We present a parallel implementation of nonlinear compact schemes with the following aims that address some of the drawbacks of existing approaches: the compact scheme does not involve any parallelization-related approximations or decoupling, the implementation is straightforward and does not require any complicated scheduling, the mathematical complexity of the

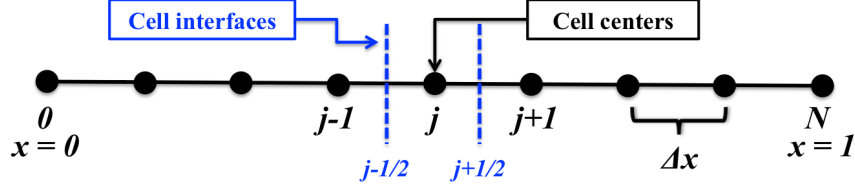


Figure 1: Schematic diagram illustrating the domain discretization.

tridiagonal solver is not significantly higher than its uniprocessor counterpart, and the overall scheme should exhibit the higher computational efficiency observed in single-processor simulations at parallelization levels of practical relevance and therefore be a viable alternative for simulations such as [9]. We realize these aims by implementing an efficient, scalable tridiagonal solver based on the partitioning or substructuring approach [19, 23, 49, 69]. Our implementation differs from other algorithms in that the reduced tridiagonal system representing the subdomain boundaries is solved iteratively. The reduced system represents the coupling between points separated by the subdomain size and is expected to be diagonally dominant. We stress here that we iterate to machine-zero accuracy, and thus our solution does not have any parallelization errors. We demonstrate our algorithm and present results with the fifth-order CRWENO scheme; however, our approach is applicable to the hybrid compact-WENO schemes as well.

This report is organized as follows. Section 2 describes the numerical method and the CRWENO scheme, as well as the numerical properties that motivate its use. Section 3 describes our parallel implementation of these algorithms and presents a scalability analysis of the proposed method for some representative, manufactured problems. Large core-count simulations of physically relevant problems are reported in Section 4. Conclusions are summarized in Section 5.

2 Compact Finite-Difference Schemes

A hyperbolic conservation law can be expressed in the differential form as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}_i(\mathbf{u})}{\partial x_i} = 0 \quad \text{in } \Omega; \quad i = 1, \dots, D, \quad (1)$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{for } x \in \Omega, \quad (2)$$

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t) \quad \text{for } x \in \partial\Omega, \quad (3)$$

where \mathbf{u} is an n -dimensional vector of conserved quantities, $\mathbf{x} = [x_i; i = 1, \dots, D]^T$ is the position vector in D -dimensional space, \mathbf{f}_i is the n -dimensional flux vector along the i th dimension, $\mathbf{u}_0(\mathbf{x})$ is the initial solution, and $\mathbf{g}(\mathbf{x}, t)$ is the boundary condition. The domain is given by Ω , and $\partial\Omega$ denotes the boundary. In the following paragraphs, the numerical discretization of the one-dimensional ($D = 1$) conservation law is considered, but the methodology can be extended trivially to multiple dimensions. Figure 1 shows a representative domain of unit length and its discretization with an $N + 1$ -point grid. A conservative, finite-difference discretization of the one-dimensional conservation law is given by

$$\frac{d\mathbf{u}_j}{dt} + \frac{1}{\Delta x} (\mathbf{h}_{j+1/2} - \mathbf{h}_{j-1/2}) = 0, \quad (4)$$

where $\mathbf{u}_j = \mathbf{u}(x_j)$; $x_j = j\Delta x$ is the cell-centered value and $\mathbf{h}_{j\pm 1/2}$ is the numerical flux at the interface. The numerical flux function $\mathbf{h}(x)$ is required to satisfy exactly

$$\left. \frac{\partial \mathbf{f}}{\partial x} \right|_{x=x_j} = \frac{1}{\Delta x} [\mathbf{h}(x_{j+1/2}, t) - \mathbf{h}(x_{j-1/2}, t)] \quad (5)$$

and can thus be defined implicitly as

$$\mathbf{f}(x) = \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} h(\xi) d\xi. \quad (6)$$

Equation (4) is exact, and no numerical approximation is involved in the discretization. It represents an ordinary differential equation (ODE) in time and can thus be solved by using an appropriate ODE solver. In the current implementation, the integrators available in the TS module of PETSc [5–7] is used to solve Eq. 4. Unless otherwise mentioned, the numerical examples in this report are solved by using the classical fourth-order, four-stage Runge-Kutta scheme.

The “reconstruction” step computes the numerical flux at the interface, $\mathbf{h}_{j+1/2}$, from the cell-centered values of the flux function, \mathbf{f}_j , to the desired order of accuracy. The approximate flux function $\hat{f}(x) \approx h(x)$ satisfies the relation

$$\begin{aligned} \left. \frac{\partial \mathbf{f}}{\partial x} \right|_{x=x_j} &= \frac{1}{\Delta x} (\mathbf{h}_{j+1/2} - \mathbf{h}_{j-1/2}) \\ &= \frac{1}{\Delta x} (\hat{\mathbf{f}}_{j+1/2} - \hat{\mathbf{f}}_{j-1/2}) + O(\Delta x^r), \end{aligned} \quad (7)$$

where r is the desired order of the scheme. The solution to hyperbolic conservation laws represents the convection or propagation of the characteristic quantities, and the reconstruction of numerical flux needs to be upwinded according to the direction of convection to yield robust solutions. In the following subsections, the left-biased reconstruction of a scalar interface flux is described, and the formulation for a right-biased reconstruction can be obtained by reflecting the expressions around the given interface. The final interface flux is obtained by appropriately combining the left- and right-biased flux based on the local characteristic behavior.

2.1 WENO Schemes

The WENO schemes [37, 45] use the idea of adaptive stenciling to achieve high order accuracy when the solution is smooth and yield non-oscillatory solutions across discontinuities. At a given interface, there are r candidate stencils for an r th-order interpolation. Optimal coefficients exist for each of these stencils such that the weighted sum results in a $(2r - 1)$ th order interpolation. Nonlinear weights are obtained by scaling these optimal weights by local smoothness of the solution such that they approach the optimal values when the solution is smooth and approach zero for stencils that contain discontinuous data. The final scheme is the weighted sum of the r th-order stencils with the nonlinear weights. In this study, we implement the fifth-order WENO scheme that is constructed by the following three third-order schemes:

$$\hat{f}_{j+1/2}^1 = \frac{1}{3}f_{j-2} - \frac{7}{6}f_{j-1} + \frac{11}{6}f_j, \quad (8)$$

$$\hat{f}_{j+1/2}^2 = -\frac{1}{6}f_{j-1} + \frac{5}{6}f_j + \frac{1}{3}f_{j+1}, \quad (9)$$

$$\hat{f}_{j+1/2}^3 = \frac{1}{3}f_j + \frac{5}{6}f_{j+1} - \frac{1}{6}f_{j+2}. \quad (10)$$

The optimal coefficients are, respectively, $c_1 = 0.1$, $c_2 = 0.6$, and $c_3 = 0.3$. Multiplying each of the third-order scheme with the corresponding optimal coefficient and adding results in the fifth-order scheme, we obtain

$$\hat{f}_{j+1/2} = \frac{1}{30}f_{j-2} - \frac{13}{60}f_{j-1} + \frac{47}{60}f_j + \frac{27}{60}f_{j+1} - \frac{1}{20}f_{j+2}. \quad (11)$$

The nonlinear weights are computed from the optimal coefficients and the local solution smoothness. The original formulation for the WENO scheme [37] defines the weights as

$$\omega_k = \frac{\alpha_k}{\sum_k \alpha_k}; \quad \alpha_k = \frac{c_k}{(\epsilon + \beta_k)^p}; \quad i = 1, \dots, 3, \quad (12)$$

where $\epsilon = 10^{-6}$ is a small number to prevent division by zero, and the smoothness indicators (β_k) for the stencils are given by

$$\beta_1 = \frac{13}{12}(f_{j-2} - 2f_{j-1} + f_j)^2 + \frac{1}{4}(f_{j-2} - 4f_{j-1} + 3f_j)^2, \quad (13)$$

$$\beta_2 = \frac{13}{12}(f_{j-1} - 2f_j + f_{j+1})^2 + \frac{1}{4}(f_{j-1} - f_{j+1})^2, \quad (14)$$

$$\text{and } \beta_3 = \frac{13}{12}(f_j - 2f_{j+1} + f_{j+2})^2 + \frac{1}{4}(3f_j - 4f_{j+1} + f_{j+2})^2. \quad (15)$$

Subsequent studies [10, 35] reported several drawbacks with this definition of the weights, including suboptimal convergence for certain classes of problems and excessive dissipation around discontinuities; and alternative definitions were proposed [10, 12, 35, 72]. Based on a detailed nonlinear analysis of the numerical properties of the WENO schemes [26], the weights defined in [72] are used in this study. The weights are defined as

$$\omega_k = \frac{\alpha_k}{\sum_k \alpha_k}; \quad \alpha_k = c_k \left[1 + \left(\frac{\tau}{\epsilon + \beta_k} \right)^p \right], \quad (16)$$

where $\tau = (f_{j-2} - 4f_{j-1} + 6f_j - 4f_{j+1} + f_{j+2})^2$. Multiplying Eqs. (8)–(10) by the nonlinear weights and then summing them results in the fifth-order WENO scheme (WENO5):

$$\hat{f}_{j+1/2} = \frac{\omega_1}{3} f_{j-2} - \frac{1}{6}(7\omega_1 + \omega_2) f_{j-1} + \frac{1}{6}(11\omega_1 + 5\omega_2 + 2\omega_3) f_j + \frac{1}{6}(2\omega_2 + 5\omega_3) f_{j+1} - \frac{\omega_3}{6} f_{j+2}. \quad (17)$$

When the solution is smooth, $\omega_k \rightarrow c_k$, and Eq. (17) reduces to Eq. (11).

2.2 CRWENO Schemes

Compact finite-difference schemes [42] compute the unknown interface flux or the flux derivative implicitly and require the solution of a system of equations. This coupling results in a higher spectral resolution, lower absolute errors, and smaller interpolation stencils, compared with standard finite-difference schemes of the same order of convergence. A general, conservative compact scheme to approximate the interface flux can be expressed as

$$A \left(\hat{f}_{j+1/2-m}, \dots, \hat{f}_{j+1/2}, \dots, \hat{f}_{j+1/2+m} \right) = B \left(f_{j-n}, \dots, f_j, \dots, f_{j+n} \right), \quad (18)$$

where A and B are linear operators, and m and n are control their stencil widths. The derivation of the CRWENO scheme follows that of the WENO scheme but with compact schemes. At a given interface, there exist r candidate r th-order compact schemes such that their weighted sum (with optimal coefficients) is a $(2r - 1)$ th-order compact scheme,

$$\begin{aligned} \sum_{k=1}^r c_k A_k^r \left(\hat{f}_{j+1/2-m}, \dots, \hat{f}_{j+1/2+m} \right) &= \sum_{k=1}^r c_k B_k^r \left(f_{j-n}, \dots, f_{j+n} \right) \\ \Rightarrow A^{2r-1} \left(\hat{f}_{j+1/2-m}, \dots, \hat{f}_{j+1/2+m} \right) &= B^{2r-1} \left(f_{j-n}, \dots, f_{j+n} \right). \end{aligned} \quad (19)$$

The optimal weights are then replaced by solution-dependent, nonlinear weights (ω_k), and the CRWENO scheme can be expressed as

$$\sum_{k=1}^r \omega_k A_k^r \left(\hat{f}_{j+1/2-m}, \dots, \hat{f}_{j+1/2+m} \right) = \sum_{k=1}^r \omega_k B_k^r \left(f_{j-n}, \dots, f_{j+n} \right). \quad (20)$$

The present implementation of the CRWENO schemes use the nonlinear weights defined for the WENO scheme in Section 2.1.

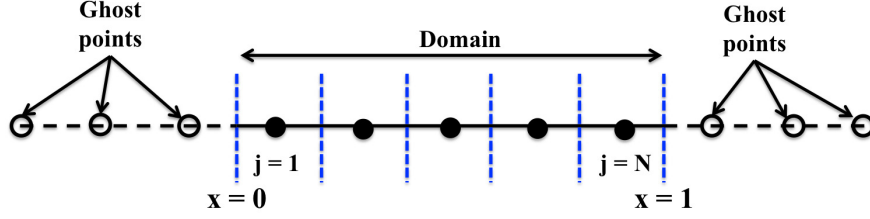


Figure 2: Boundary treatment using ghost points.

The fifth-order CRWENO scheme [25,27] is constructed by combining three third-order compact schemes. The candidate compact schemes and their optimal coefficients are given by

$$\frac{2}{3}\hat{f}_{j-1/2} + \frac{1}{3}\hat{f}_{j+1/2} = \frac{1}{6}(f_{j-1} + 5f_j); \quad c_1 = \frac{2}{10}, \quad (21)$$

$$\frac{1}{3}\hat{f}_{j-1/2} + \frac{2}{3}\hat{f}_{j+1/2} = \frac{1}{6}(5f_j + f_{j+1}); \quad c_2 = \frac{5}{10}, \quad (22)$$

$$\frac{2}{3}\hat{f}_{j+1/2} + \frac{1}{3}\hat{f}_{j+3/2} = \frac{1}{6}(f_j + 5f_{j+1}); \quad c_3 = \frac{3}{10}. \quad (23)$$

The fifth-order accurate compact scheme is obtained by multiplying the third-order schemes by their optimal coefficients (c_k) and adding

$$\frac{3}{10}\hat{f}_{j-1/2} + \frac{6}{10}\hat{f}_{j+1/2} + \frac{1}{10}\hat{f}_{j+3/2} = \frac{1}{30}f_{j-1} + \frac{19}{30}f_j + \frac{1}{3}f_{j+1}. \quad (24)$$

The optimal coefficients are replaced with the nonlinear weights ω_k , and we get the fifth-order CRWENO scheme (CRWENO5),

$$\begin{aligned} \left(\frac{2}{3}\omega_1 + \frac{1}{3}\omega_2\right)\hat{f}_{j-1/2} + \left[\frac{1}{3}\omega_1 + \frac{2}{3}(\omega_2 + \omega_3)\right]\hat{f}_{j+1/2} + \frac{1}{3}\omega_3\hat{f}_{j+3/2} \\ = \frac{\omega_1}{6}f_{j-1} + \frac{5(\omega_1 + \omega_2) + \omega_3}{6}f_j + \frac{\omega_2 + 5\omega_3}{6}f_{j+1}. \end{aligned} \quad (25)$$

The weights ω_k are computed by Eq. (16) and Eqs. (13)–(15). When the solution is smooth, $\omega_k \rightarrow c_k$, and Eq. (25) reduces to (24).

Equation (25) requires the solution of a tridiagonal system of equations. Since the weights ω_k are solution-dependent, the system of equations has to be solved along each grid line at every time-integration step/stage.

2.3 Boundary Treatment

The implementation of the scheme at the physical boundaries on a bounded domain is critical to the accuracy and stability of the overall scheme. In this study, the physical domain is extended by using “ghost” points, as shown in Fig. 2. The dependent variables at the ghost points are set such that the interface flux is consistent with the physical boundary conditions. Standard finite-difference schemes can thus be applied without any modifications at the boundary interfaces. However, the application of compact schemes is not possible because of the absence of “ghost interfaces.” Specification of the flux at interfaces outside the physical domain may not be possible, except for very simple problems.

The implementation of the CRWENO5 scheme [26,27] used the fifth-order WENO scheme at the boundary interfaces, and a numerical analysis of the overall discretization [25] showed that this boundary treatment

was numerically stable. In this study, we use this implementation, expressed as follows:

$$\begin{aligned} j = 0 & : \text{Eq. (17);} \\ j = 1, \dots, N-1 & : \text{Eq. (25);} \\ j = N & : \text{Eq. (17).} \end{aligned}$$

The overall scheme needs three ghost points at the boundaries, and the resulting tridiagonal system of equations along a grid line has the first and last diagonal element as unity and off-diagonal elements as zero.

2.4 Numerical Properties

The numerical properties of the fifth-order compact finite-differencing schemes are briefly discussed in this section. More detailed discussions have been previously presented [25–27] that demonstrate the superior numerical properties of the CRWENO5 scheme compared with the WENO5 scheme. Equation (24) is the linear fifth-order compact scheme underlying the CRWENO5 scheme. The resulting approximation for the first derivative of the flux function can be expressed as

$$\begin{aligned} \frac{3}{10}f_{x,j-1} + \frac{6}{10}f_{x,j} + \frac{1}{10}f_{x,j+1} &= \frac{1}{\Delta x} \left(\frac{-1}{30}f_{j-2} - \frac{18}{30}f_{j-1} + \frac{9}{30}f_j + \frac{10}{30}f_{j+1} \right) \\ \Rightarrow f_{x,j} &= f_{\Delta,j} + \frac{1}{600} \left. \frac{\partial^6 f}{\partial x^6} \right|_j \Delta x^5 + \frac{1}{2100} \left. \frac{\partial^7 f}{\partial x^7} \right|_j \Delta x^6 + O(\Delta x^7), \end{aligned} \quad (26)$$

where the term f_{Δ} denotes the finite-difference approximation to the first derivative. The corresponding expression for Eq. (11), which is the underlying linear interpolation for the WENO5 scheme, can be expressed as

$$\begin{aligned} f_{x,j} &= \frac{1}{\Delta x} \left(\frac{-1}{30}f_{j-3} + \frac{1}{4}f_{j-2} - f_{j-1} + \frac{1}{3}f_j + \frac{1}{2}f_{j+1} - \frac{1}{20}f_{j+2} \right) \\ &+ \frac{1}{60} \left. \frac{\partial^6 f}{\partial x^6} \right|_j \Delta x^5 + \frac{1}{140} \left. \frac{\partial^7 f}{\partial x^7} \right|_j \Delta x^6 + O(\Delta x^7). \end{aligned} \quad (27)$$

Examination of the leading-order dissipation and dispersion error terms shows that the compact interpolation scheme yields solutions with 1/10 the dissipation error and 1/15 the dispersion error of the solutions obtained by the noncompact scheme, for the same order of convergence. This implies that for smooth solutions, the fifth-order WENO scheme requires $10^{1/5}$ or approximately 1.5 times more grid points per dimension to yield a solution of comparable accuracy as the fifth-order CRWENO scheme. This fact will be used in later sections when comparing the computational efficiencies of the CRWENO5 and WENO5 schemes on multiple processors. The numerical cost of the CRWENO5 scheme and its tridiagonal solver is less than the cost of the WENO5 scheme on the finer grid (especially so in multiple dimensions, where the grid needs to be refined in each dimension) for solutions obtained on single processors [27].

The primary motivation for the use of compact schemes is the high spectral resolution that results in more accurate modeling of moderate and small length scales. Compact schemes (and other high-resolution schemes) are thus well suited for applications with a large range of length scales. The spectral properties of finite-difference schemes are quantified by a Fourier analysis, and a detailed linear analysis as well as a nonlinear spectral analysis of the CRWENO5 scheme has been presented [26, 27]. Figure 3 shows the dispersion and dissipation properties of the CRWENO5, WENO5, and their underlying linear schemes. The spectral properties of the linear schemes, Eqs. (11) and (24), are obtained through a Fourier analysis of the finite-difference approximation. The spectral properties of the WENO5 and CRWENO5 schemes, Eqs. (17) and (25), are obtained using a nonlinear spectral analysis [22, 26]. The linear fifth-order compact and the CRWENO5 schemes have significantly higher spectral resolution than do the corresponding standard fifth-order and WENO5 schemes respectively. The compact schemes also exhibit lower dissipation for the low and moderate wavenumbers that are accurately modeled, while they show higher dissipation for very

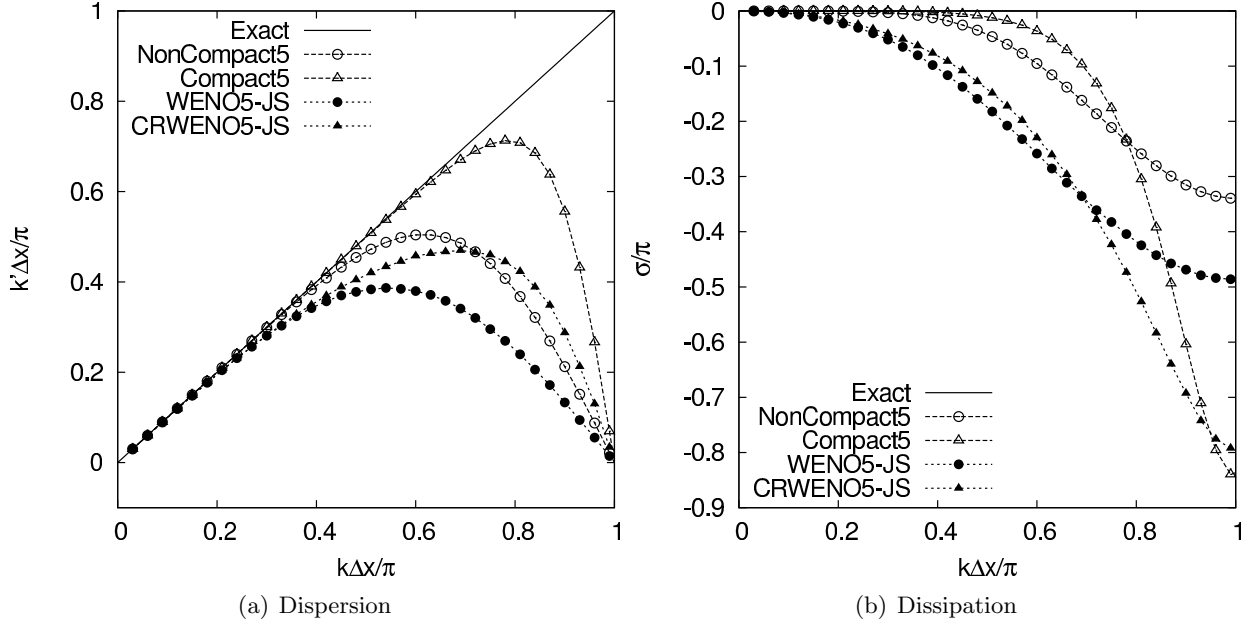


Figure 3: Spectral properties of the linear and nonlinear schemes: “Compact5” refers to Eq. (24); “NonCompact5” refers to Eq. (11); CRWENO5 and WENO5 refer to Eqs. (25) and (17), respectively.

high wavenumbers that are incorrectly aliased to lower wavenumbers. Thus, this higher dissipation reduces the small length-scale errors. Moreover, the nonlinear weights contribute significantly to the degradation of spectral resolution and increase the dissipation of the schemes, for both the CRWENO5 and WENO5 schemes (compared with their linear counterparts).

Table 1 compares the bandwidth resolving efficiency [42] of Eq. (24) with standard finite-difference schemes as well as other high-resolution schemes in the literature. The resolving efficiency of the fifth-order compact scheme is higher than that of the standard fifth-, seventh-, and ninth-order finite-difference schemes. It compares well with the sixth- and eighth-order central schemes proposed in [42], as well as the bandwidth-optimized WENO schemes proposed in [47].

3 Parallel Implementation

The fifth-order CRWENO scheme described in the previous section, Eq. (25), results in a solution-dependent, tridiagonal system of equations of the form

$$A\hat{\mathbf{f}} = \mathbf{r}; \text{ where } \mathbf{r} = B\mathbf{f} + \mathbf{b}. \quad (28)$$

Referring to the discretized domain shown in Fig. 1 with $N - 1$ interior grid points $(1, \dots, N - 1)$, and N interfaces, the tridiagonal, left-hand side matrix A is of size $N \times N$, $\hat{\mathbf{f}} = [\hat{f}_{j+1/2}; j = 0, \dots, N - 1]^T$ is the N -dimensional vector of unknown flux at the interfaces, B is a $N \times (N - 1)$ matrix representing the right-hand side operator, and $\mathbf{f} = [f_j; j = 1, \dots, N - 1]^T$ is the vector of (known) flux at the cell centers. The boundary terms are represented by \mathbf{b} .

Parallel implementations of the CRWENO scheme (as well as the hybrid compact-ENO/WENO schemes) depend on the efficient, multiprocessor solution of Eq. (28). This section describes our implementation of a parallel tridiagonal solver and demonstrates its performance and scalability.

Table 1: Bandwidth resolving efficiencies for various schemes for an error tolerance of 0.01

Scheme	Bandwidth Resolving Efficiency
5th-order standard Eq. (11)	0.35
7th-order standard	0.42
9th-order standard	0.48
5th-order compact Eq. (24)	0.61
WENO-SYMBO ($r = 3$) [47]	0.49
WENO-SYMBO ($r = 4$) [47]	0.56
6th-order central compact (tridiagonal) [42]	0.50
8th-order central compact (tridiagonal) [42]	0.58

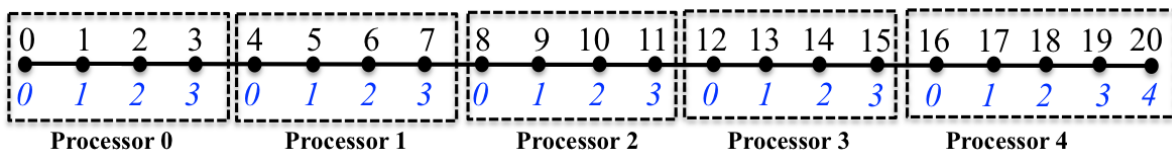


Figure 4: Example of a partitioned grid: 21 points distributed among 5 processors; the black numbers indicate the global numbering of points, while the blue numbering is the local numbering on each processor.

3.1 Tridiagonal Solver

We implement a parallel tridiagonal solver with the following aims: it solves the system to sufficient accuracy such that no parallelization-induced errors are introduced into the numerical scheme, it does not significantly increase the mathematical complexity with respect to the Thomas algorithm on a single processor, and it does not involve any collective communications. We use the substructuring or partitioning approach for the underlying algorithm. Figure 4 shows an example of a case where a 21-point grid is distributed on five processors. The global and local numbering of the points are shown. We explain the substructuring process by reordering the tridiagonal system as shown in Fig. 5. The rows corresponding to the first point on each subdomain (except the first, which is a physical boundary point) are placed at the bottom of the matrix in the order of the processor rank on which they reside. These rows are marked by the dotted box in Fig. 5. This results in decoupled tridiagonal blocks on each processor as marked by the solid boxes in the same figure. The solution is then obtained as follows:

- Stage 1 – Parallel elimination of the tridiagonal blocks on each processor comprising all points of the subdomain except the first point (unless it's the first global point, i.e., a physical boundary): This stage requires no communication. The resultant matrix is shown in Fig. 6. New nonzeros are created (\hat{a}_i), which are stored in the same memory locations as a_i that are eliminated. Note that each row eliminates the one after it as well as the first local row that was reordered; for example, row 5 eliminates rows 6 and 4, after which row 6 eliminates rows 7 and 4, and so on.
- Stage 2 – Elimination of the first row on each processor (except the first) using the last row of the previous processor: This stage requires a one-way communication—each processor, except the last, sends its last row to the next processor. After receiving this data, each processor then eliminates the subdiagonal term (a_i) of the first local row, i.e., the rows that were reordered, using the row received. The superdiagonal term (\hat{c}_i) of the first local row is eliminated using the last local row. For example, processor 1 sends row 7 to processor 2. Processor 2 then eliminates the subdiagonal term (a_8) of row 8

b_0	c_0															
	\hat{b}_1	c_1														
		\hat{b}_2	c_2													
			\hat{b}_3					c_3								
			b_5	c_5						a_5						
				\hat{b}_6	c_6				\hat{a}_6							
					\hat{b}_7			\hat{a}_7	c_7							
					b_9	c_9				a_9						
						\hat{b}_{10}	c_{10}			\hat{a}_{10}						
							\hat{b}_{11}			\hat{a}_{11}	c_{11}					
							b_{13}	c_{13}			a_{13}					
Processor 3								\hat{b}_{14}	c_{14}			\hat{a}_{14}				
									\hat{b}_{15}			\hat{a}_{15}	c_{15}			
										b_{17}	c_{17}			a_{17}		
Processor 4											\hat{b}_{18}	c_{18}			\hat{a}_{18}	
												\hat{b}_{19}	c_{19}			\hat{a}_{19}
													\hat{b}_{20}			\hat{a}_{20}
		a_4			\hat{c}_4					\hat{b}_4						
				a_8			\hat{c}_8				\hat{b}_8					
					a_{12}			\hat{c}_{12}			\hat{b}_{12}					
						a_{16}			\hat{c}_{16}			\hat{b}_{16}				

Figure 6: Tridiagonal matrix after Stage 1: The hat indicates values altered during the elimination.

gather-and-solve method where the entire reduced system is gathered on one processor, solved, and the solution scattered back. Multidimensional problems require the solution to a system of equations along each grid line, and the solutions to several systems are required at each iteration. The efficiency of the latter approach can be improved by gathering the reduced systems for different grid lines on different processors and solving them simultaneously. One can observe that both approaches have significant communication overhead for this application, especially since each processor has only one row of the reduced system. The overall algorithm is not very scalable because the cost of the third stage increases significantly as the number of processors increases for a given size of the system. This will be demonstrated shortly.

Our approach stems from the observation that the tridiagonal system of equations results from a finite-difference approximation of a hyperbolic flux term. Although the compact scheme couples the computation of the flux at neighboring interfaces, the coupling between interfaces that are several grid points away will be very weak. The reduced system of equations represents the coupling between the first grid points of each subdomain and are separated by the subdomain size. This system will have very strong diagonal dominance for $p \ll N$. As $p \rightarrow O(N)$, the diagonal dominance will decrease. We thus propose solving the reduced system using Jacobi iterations with an initial guess that is the solution of the corresponding diagonal system (i.e., neglecting the off-diagonal terms). For large subdomain sizes (low number of processors), the error in

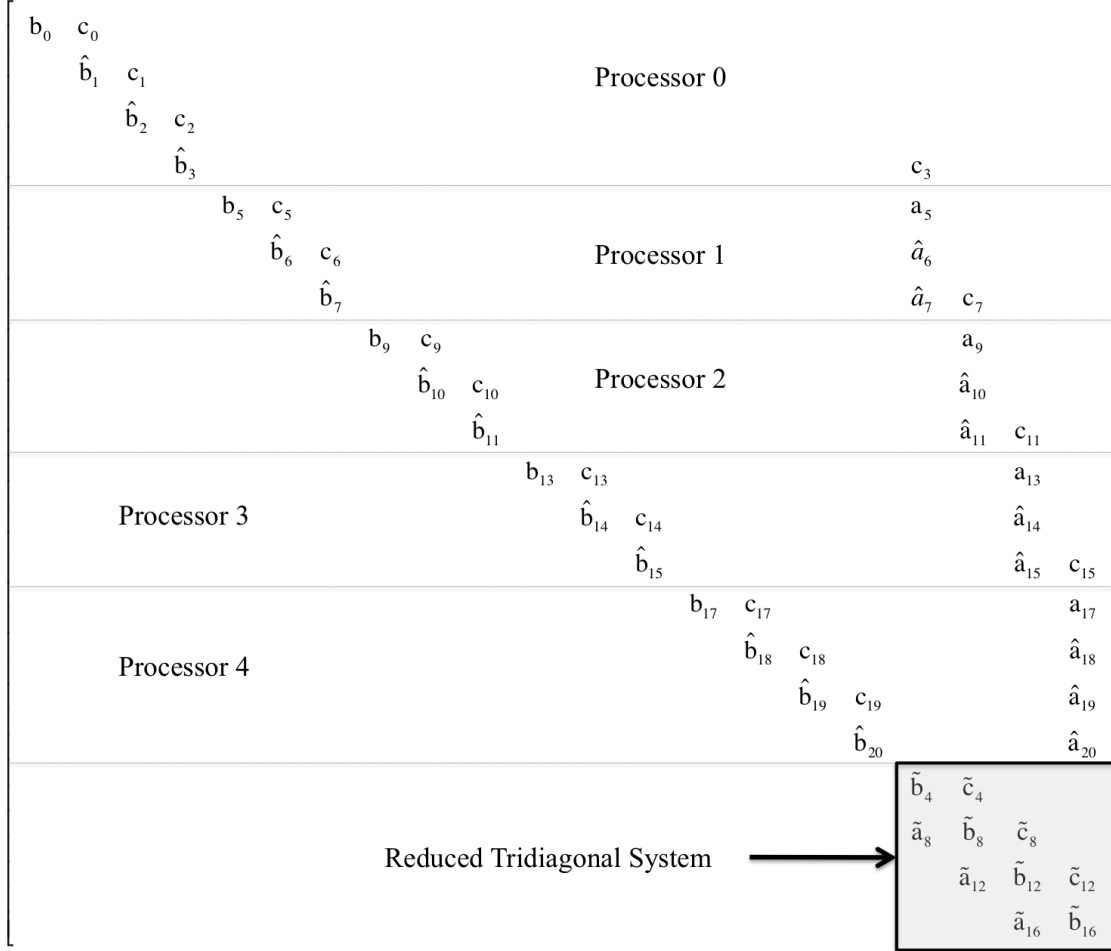


Figure 7: Tridiagonal matrix after Stage 2: Elimination of all the rows results in a reduced tridiagonal equation (shown by the box), with each row residing on a different processor.

the initial guess is expected to be below machine zero, and no iterations will be needed. As the number of processors increases for the same problem size, the diagonal dominance will decrease, and the number of iterations needed to converge to machine zero accuracy will increase steadily.

We note here that although the reduced system is solved iteratively, it is solved to machine-zero accuracy in all our computations. This ensures that the tridiagonal solver does not introduce any approximations in the overall algorithm, and in all our numerical experiments we verify that the solutions obtained on multiple processors are identical to those obtained on a single processor. We further reduce the communication overhead of our algorithm by eliminating tolerance-based exit criteria for the Jacobi iterations and using a specified number of iterations. Thus, no collective communications are required, and each iteration requires only point-to-point communications between neighboring processors. However, this does require an a priori estimate of the number of iterations required to ensure machine-zero convergence of the reduced system.

We present an example to illustrate the behavior of the tridiagonal solver. We consider the tridiagonal system corresponding to Eq. (24) with $N = 1024$ and a random right-hand side. Figure 8 shows the elements of the $p/2$ th column of the inverse of the reduced system for 16, 64, 128, and 256 processors (where p is the number of processors), corresponding to subdomain sizes of 64, 16, 8, and 4 points, respectively. The y -axis

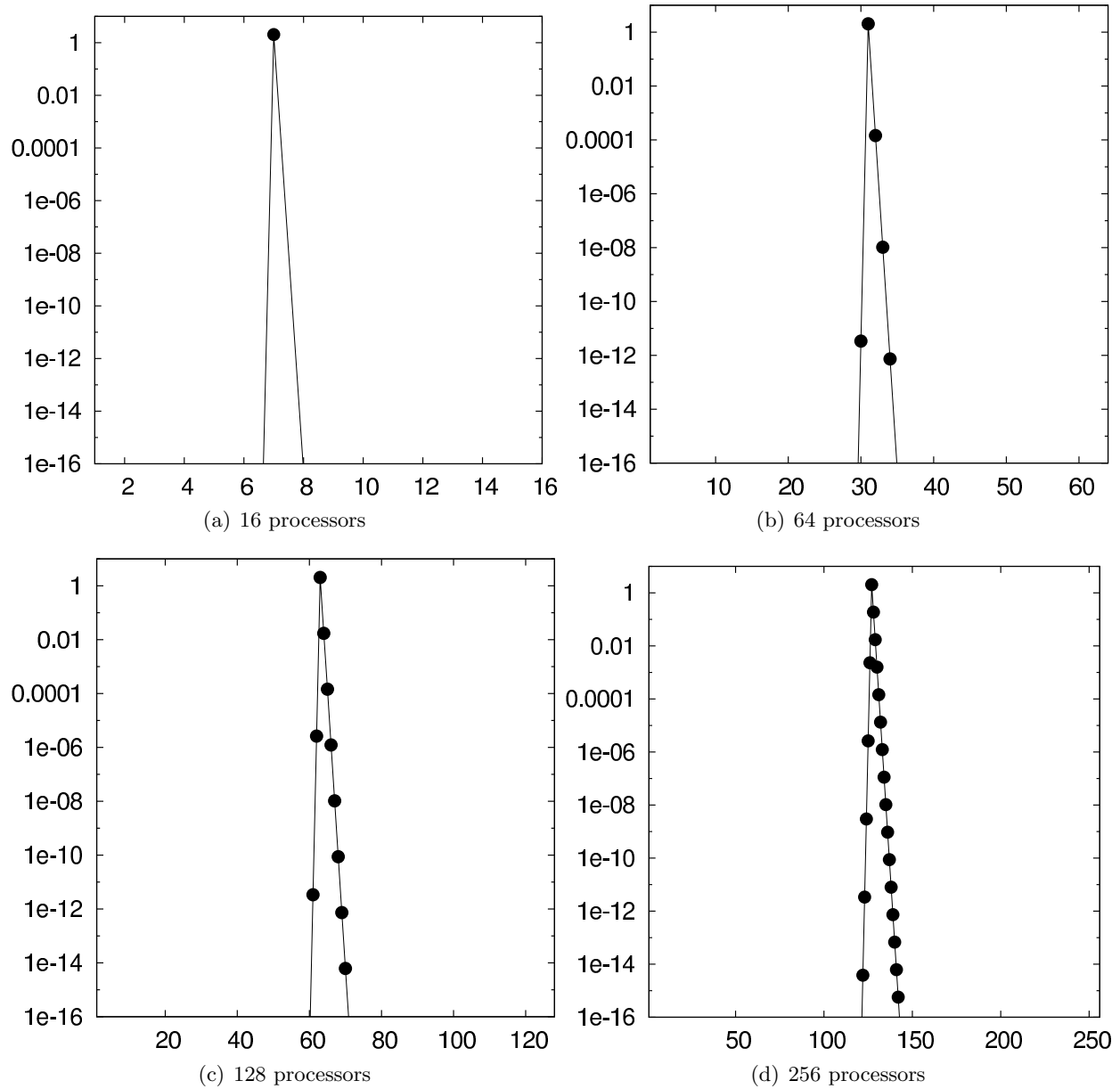


Figure 8: Elements of the $p/2$ th column of the inverse of the reduced system for various number of processors (p).

range is set to show only the elements larger than machine zero (10^{-16}). We observe that for a subdomain size of 64 points (16 processors), only the diagonal element has a value higher than machine-zero. The reduced system and its inverse can thus be treated as diagonal matrices, and no Jacobi iterations are needed to solve it. The number of non-machine-zero elements grows and the diagonal dominance decreases as the subdomain size decreases; consequently, Jacobi iterations are needed to solve the reduced system. We note that for this example even for a subdomain size of 4 points (256 processors), the diagonal element is an order of magnitude larger than the largest off-diagonal element.

Figure 9 shows the error in the solution of the tridiagonal system ($N = 65536$) as a function of the subdomain size, with various numbers of Jacobi iterations as well as the direct solution to the reduced system using the gather-and-solve approach. We observe that for a given number of Jacobi iterations, the tridiagonal solver yields a solution identical to the direct solution until a certain subdomain size; for smaller subdomains, the error gradually grows. As an example, with zero Jacobi iterations specified, the iterative solver yields solutions identical to the direct solver for subdomain sizes larger than 64 points per processor. Two Jacobi iterations are able to yield solutions without errors for subdomain sizes larger than 32 points per processor. As the subdomain size shrinks, we need a larger number of Jacobi iterations to ensure that the solution is identical to that of a direct method. We note, however, that the iterative solution need not be identical to the direct solution; it is sufficient if the error in the iterative solution is below machine zero. In this example, 6 Jacobi iterations are thus sufficient for a subdomain size of 4 points per processor to ensure that the overall scheme has no parallelization-induced errors.

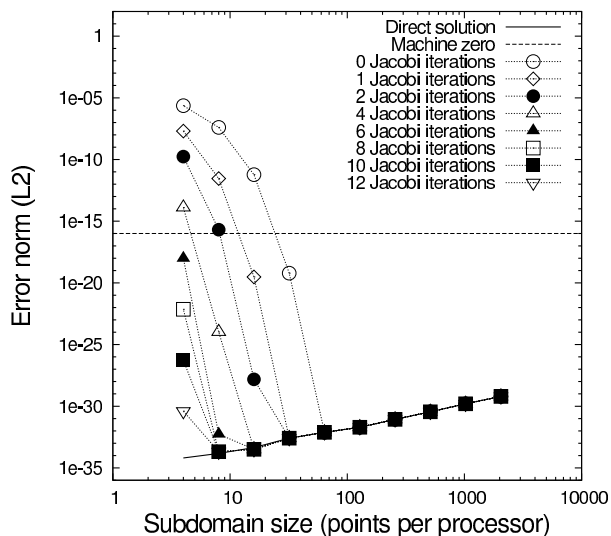


Figure 9: Tridiagonal solver: error as a function of subdomain size.

3.2 Performance Analysis

We analyze the performance of the nonlinear compact scheme with our implementation of the tridiagonal solver in this section. We apply the numerical method described in Section 2 to the inviscid Euler equations [39]. The scalar reconstruction schemes are applied to each component of the vector quantities. We investigate the computational efficiency of the CRWENO5 with respect to the WENO5 scheme for multi-processor solutions. It has been previously demonstrated that the CRWENO5 scheme is more efficient than the WENO5 scheme for solutions on a single processor [25, 27]. While the cost of the CRWENO5 scheme is higher than that of the WENO5 scheme on the same grid (because of the tridiagonal system solution), the former yields solutions with higher accuracy and resolution than does the latter; therefore, the WENO5 scheme requires a finer grid (consequently higher numerical cost) to yield comparable solutions. Our implementation of the parallel tridiagonal solver is such that its relative cost increases as the number of processors increases for the same domain size, because of an increase in the number of Jacobi iterations needed for machine-zero convergence of the reduced system. It is thus expected that there will be a critical sub-domain size beyond which (i.e., for finer partitioning) the increasing cost of the tridiagonal solver will cause the efficiency of the CRWENO5 scheme to be lower than that of the WENO5 scheme. In other words, it will be possible to obtain comparable solutions with the WENO5 scheme on a finer grid at a lower numerical cost. Our objective is to find this critical subdomain size through representative numerical experiments and demonstrate that compact schemes retain their higher efficiency for domain partitioning sizes of practical relevance.

All computations presented in this section are carried out on Vesta, a Blue Gene/Q development rack maintained by the Argonne Leadership Computing Facility (ALCF) [2]. The system has two racks, each with 10, 24 compute nodes. Each compute node has a 1600 MHz PowerPC A2 processor with a 16-core chip and 16 GB RAM. There are thus 327, 68 cores with a peak performance of 419.44 teraflops, and our choice of test cases presented below is based on these available resources.

Our numerical experiments involve two types of problems, both involving the advection of density waves on periodic domains—a smooth sinusoidal wave and a wave that is a sum of sinusoidal waves with all the

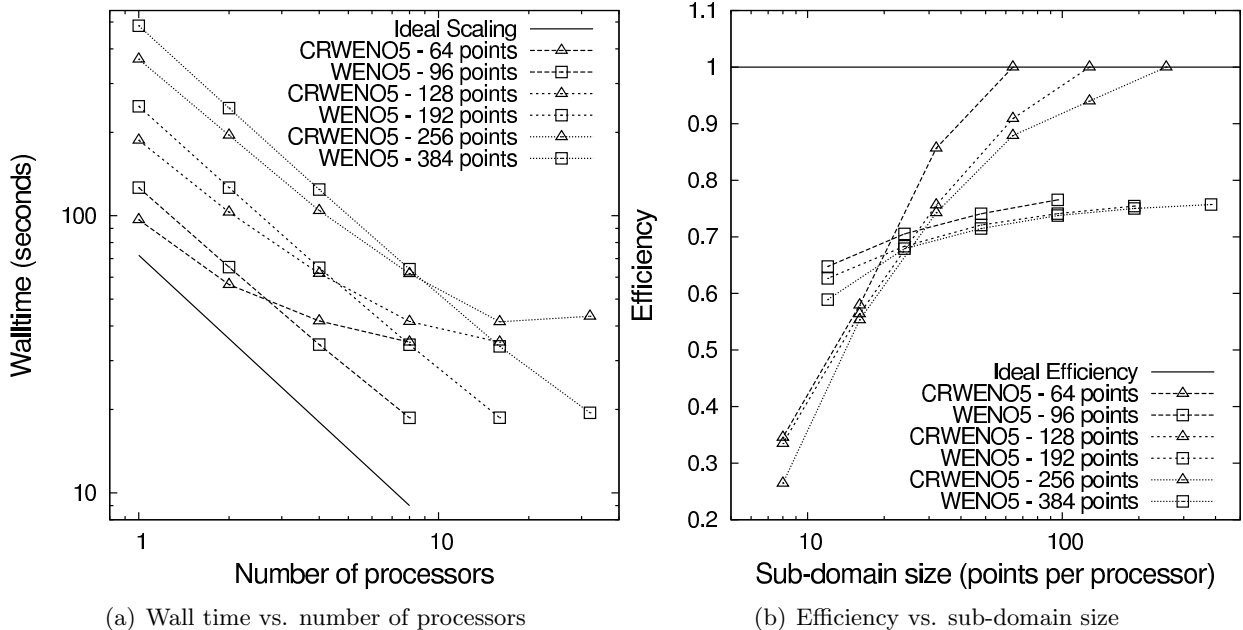


Figure 10: One-dimensional advection of density sine wave – Wall times and efficiencies for the WENO5 and CRWENO5 schemes on grids with 64, 96, 128, 192, 256, and 384 points (data in Table 2).

wavenumbers supported by the grid. We study these problems in one, two, and three spatial dimensions. We start with the one-dimensional advection of a density wave. The initial density, velocity, and pressure are given by

$$\begin{aligned}\rho &= \rho_0 + \tilde{\rho} \sin(2\pi x), \\ u &= 1, \quad p = 1/\gamma,\end{aligned}$$

respectively, on a unit periodic domain $x \in [0, 1]$. The specific heat ratio is $\gamma = 1.4$. The mean density ρ_0 is 1 and the amplitude of the sine wave is taken as $\tilde{\rho} = 0.1$. Solutions are obtained with the WENO5 and CRWENO5 schemes on grids with 64, 128, and 256 points (baseline grids). In addition, solutions are obtained with the WENO5 scheme on grids with 1.5 times as many points (96, 192, and 384 points). A small time step size of 10^{-4} is taken such that the errors due to the numerical time integration are negligible (relative to those from the spatial discretization). The solutions are obtained after one cycle over the periodic domain.

Table 2 shows the L_2 norm of the numerical errors and the wall times for the solutions corresponding to different grid sizes (N_{global}) and subdomain sizes (N_{local}). The number of Jacobi iterations (N_{Jac}) required to solve the reduced tridiagonal system (Section 3.1) for the CRWENO5 scheme is also shown for each solution. The WENO5 and CRWENO5 schemes both show fifth-order convergence; and, as expected from the Taylor series analysis (in Section 2.4), the CRWENO5 scheme yields solutions with errors that are an order of magnitude lower compared to the WENO5 scheme on the same grid. The WENO5 scheme yields solutions of comparable accuracy on grids that are 1.5 times finer, namely, the grids with 96, 192, and 384 points. It is verified that the numerical errors for the solutions obtained with the CRWENO5 scheme are identical for all the subdomain sizes considered for a given grid size. This verification is crucial in demonstrating that our algorithm does not in fact introduce any parallelization-induced errors. The required number of Jacobi iterations increases as the subdomain size decreases, as expected from the earlier discussion of the nature of the reduced tridiagonal system.

We compare the wall times for all the cases in order to assess the relative efficiency of the CRWENO5

Table 2: Errors (L_2) and wall times (in seconds) for the one-dimensional advection of a sinusoidal density wave.

N_{global}	N_{local}	WENO5		CRWENO5		N_{Jac}
		Error	Wall time	Error	Wall time	
1 processor						
64	64	1.0118E-07	8.5763E+01	1.1561E-08	9.6688E+01	0
96	96	1.3332E-08	1.2633E+02	-	-	-
128	128	3.1644E-09	1.6725E+02	3.3927E-10	1.8726E+02	0
192	192	4.1680E-10	2.4820E+02	-	-	-
256	256	9.8914E-11	3.2760E+02	1.0253E-11	3.6687E+02	0
384	384	1.3024E-11	4.8451E+02	-	-	-
2 processors						
64	32	1.0118E-07	4.4940E+01	1.1561E-08	5.6432E+01	0
96	48	1.3332E-08	6.5250E+01	-	-	-
128	64	3.1644E-09	8.5767E+01	3.3927E-10	1.0300E+02	0
192	96	4.1680E-10	1.2635E+02	-	-	-
256	128	9.8914E-11	1.6611E+02	1.0253E-11	1.9515E+02	0
384	192	1.3024E-11	2.4452E+02	-	-	-
4 processors						
64	16	1.0118E-07	2.4062E+01	1.1561E-08	4.1722E+01	2
96	24	1.3332E-08	3.4272E+01	-	-	-
128	32	3.1644E-09	4.4596E+01	3.3927E-10	6.1905E+01	1
192	48	4.1680E-10	6.4947E+01	-	-	-
256	64	9.8914E-11	8.4958E+01	1.0253E-11	1.0434E+02	0
384	96	1.3024E-11	1.2436E+02	-	-	-
8 processors						
64	8	1.0118E-07	1.3524E+01	1.1561E-08	3.4970E+01	4
96	12	1.3332E-08	1.8661E+01	-	-	-
128	16	3.1644E-09	2.3931E+01	3.3927E-10	4.1513E+01	2
192	24	4.1680E-10	3.4254E+01	-	-	-
256	32	9.8914E-11	4.4308E+01	1.0253E-11	6.1754E+01	1
384	48	1.3024E-11	6.4167E+01	-	-	-
16 processors						
128	8	3.1644E-09	1.3464E+01	3.3927E-10	3.4969E+01	4
192	12	4.1680E-10	1.8681E+01	-	-	-
256	16	9.8914E-11	2.3752E+01	1.0253E-11	4.1439E+01	2
384	24	1.3024E-11	3.3763E+01	-	-	-
32 processors						
256	8	9.8914E-11	1.4207E+01	1.0253E-11	4.3370E+01	6
384	12	1.3024E-11	1.9461E+01	-	-	-

scheme compared with the WENO5 scheme. The wall times for the CRWENO5 scheme on the baseline grids are compared with those of the WENO5 scheme on the 1.5x finer grids, since the errors for these cases are comparable. The WENO5 cases on 1.5x grids are run on the same number of processors (and not 1.5 times as

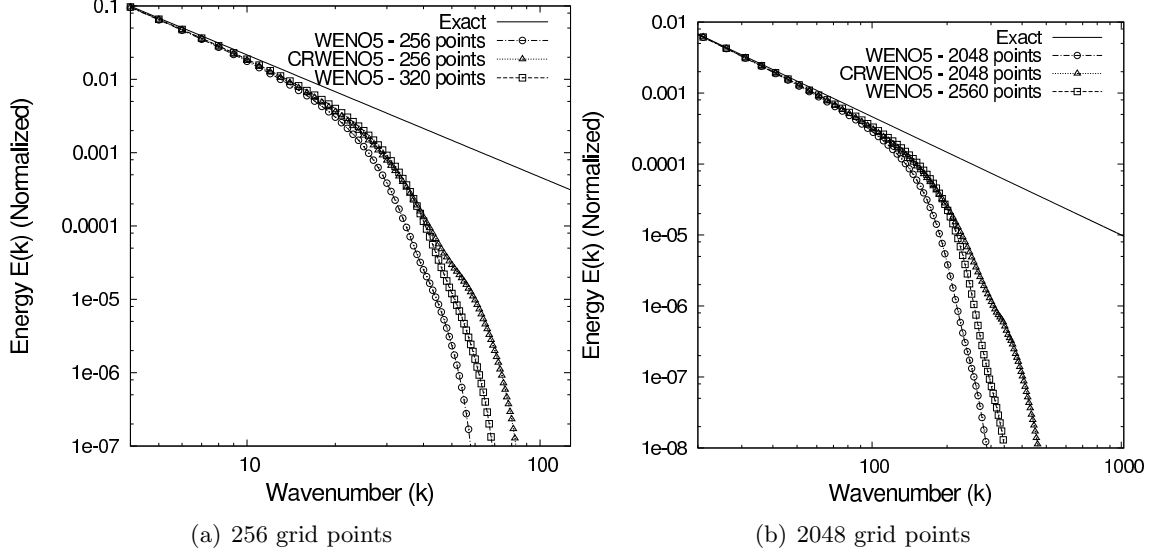


Figure 11: Energy spectrum of the numerical solutions to the one-dimensional advection of density fluctuations.

many processors) since the number of processors is usually constrained. In other words, with a given number of processors, we investigate whether the CRWENO5 scheme is more efficient, that is, whether it takes less time than the WENO5 scheme to yield a comparable solution. The solutions obtained on one processor show that the CRWENO5 scheme is more efficient; it has a lower wall time on the 64, 128, and 256 point grids than does the WENO5 scheme on the 96, 192, and 384 point grids, respectively (as demonstrated previously [27]).

As we reduce the subdomain sizes for a given case of grid sizes (e.g., CRWENO5 on the 64-points grid and WENO5 on the 96-points grid), we observe that the relative cost of the CRWENO5 scheme increases because of the increasing number of Jacobi iterations. This results in the WENO5 scheme being more efficient at smaller subdomain sizes. For example, for the CRWENO5 on the 64-points grid and the WENO5 on the 96-points grid, the CRWENO5 is faster on 1 and 2 processors (corresponding to subdomain sizes of 64 and 32) but slower for 4 and 8 processors (corresponding to subdomain sizes of 16 and 8). Similarly, for the CRWENO5 on the 128-points grid and the WENO5 on the 192-points grid, the CRWENO5 is faster on 1, 2, and 4 processors (subdomain sizes of 128, 64, and 32) but more expensive on larger numbers of processors. The CRWENO5 scheme on the 256-point grid is less expensive than the WENO5 scheme on the 384-point grid on 1, 2, 4, and 8 processors (subdomain sizes of 256, 128, 64, and 32). Figure 10(a) shows the wall time (in seconds) as a function of the number of processors for the CRWENO5 and WENO5 cases with comparable accuracy in Table 2 and shows the cost of the CRWENO5 scheme exceeding that of the corresponding WENO5 scheme as the number of processor increases for a given grid size. Figure 10(b) shows the efficiency (calculated based on the single-processor wall time of the CRWENO5 scheme on the corresponding grid) as a function of the subdomain sizes. One can observe that at larger subdomain sizes, the CRWENO5 scheme is more efficient; as the subdomain size decreases, the efficiency of the CRWENO5 scheme decreases and is lower than that of the WENO5 scheme. It can also be observed that the critical sub-domain size at which the CRWENO5 scheme ceases to be more efficient is insensitive to the global grid size. This is an important inference because it indicates that conclusions on the critical sub-domain size are expected to hold true when this algorithm is applied to larger problems on larger numbers of processors.

The conclusions in the preceding paragraphs are drawn from the solution to a smooth sine wave for which the CRWENO5 scheme reduces to a linear, fifth-order compact scheme, Eq. (24). We now consider the advection of a periodic density fluctuations comprising sine waves with all grid-supported wavenumbers. The

initial density is given by

$$\rho = \rho_0 + \tilde{\rho} \sum_{k=1}^{N/2} A(k) \sin(2\pi kx); \quad A(k) = k^{-5/6}, \quad (29)$$

with $\rho_0 = 1$, $\tilde{\rho} = 10^{-2}$, and N being the grid size. Uniform initial velocity ($u = 1$) and pressure ($p = 1/\gamma$) are prescribed. The amplitude decay ($A(k)$) is such that the energy spectrum is representative of that in turbulent flows. Although the solution is smooth, the presence of high wavenumbers results in sharp gradients that are treated as discontinuities by the WENO5 and CRWENO5 schemes. Improvement in the behavior of these schemes for these small length scales is an area of active research [26]; the current implementations do not achieve fifth-order accuracy and convergence for such solutions. However, the CRWENO5 schemes yield solutions with a higher spectral resolution than the WENO5 scheme [25, 26] on identical grids; that is, the WENO5 scheme needs a larger number of grid points to yield solutions of the same resolution. We thus compare the numerical cost of these schemes on multiple processors for solutions with comparable spectral resolutions.

Table 3: Wall times (in seconds) for the one-dimensional advection of density fluctuations.

N_{proc}	N_{local}	Wall Times			N_{Jac} CRWENO5
		WENO5	WENO5(F)	CRWENO5	
256 grid points					
1	256 (320)	1.6923E+02	2.1038E+02	1.8978E+02	0
2	128 (160)	8.6635E+01	1.0731E+02	1.0082E+02	0
4	64 (80)	4.5017E+01	5.5452E+01	5.7351E+01	2
8	32 (40)	2.3889E+01	2.9203E+01	3.3570E+01	2
16	16 (20)	1.3067E+01	1.5804E+01	2.6229E+01	6
2048 grid points					
4	512 (640)	3.3216E+03	4.1357E+03	3.8087E+03	0
8	256 (320)	1.6836E+03	2.0940E+03	1.9465E+03	0
16	128 (160)	8.6243E+02	1.0682E+03	1.0467E+03	2
32	64 (80)	4.7498E+02	5.8497E+02	6.0045E+02	2
64	32 (40)	2.5312E+02	3.1040E+02	3.8539E+02	4
128	16 (20)	1.3726E+02	1.6694E+02	2.8026E+02	6

Number in parentheses is the N_{local} for the WENO5 scheme on the finer grid (“WENO5(F)”)

We solve the problem on two different grid resolutions to show that the critical subdomain size (below which the compact schemes are less efficient) is not a function of the total problem size. We consider grids with 256 and 2048 points (with time-step sizes of 2×10^{-4} and 2×10^{-5} , respectively). Solutions are obtained after one time period. Figure 11 shows the energy spectrum of the numerical solutions obtained by the WENO5 and CRWENO5 schemes. One can see that the CRWENO5 scheme yields solutions with a higher spectral resolution at moderate and high wavenumbers than does the WENO5 on the same grid; the WENO5 scheme yields solutions with a comparable resolution on grids that have 1.25 times more points (this is a result of observation). We thus compare the numerical cost of the CRWENO5 scheme with that of the WENO5 scheme on the grid with 1.25 times more points, for both the problem sizes considered.

Table 3 shows the wall times for the WENO5 and CRWENO5 schemes for the two problem sizes (baseline grids with 256 and 2048 points) obtained on N_{proc} processors with subdomain sizes N_{local} . The column “WENO5(F)” shows the wall times for the WENO5 scheme on a grid with 1.25 times as many points as the corresponding baseline grids (grids with 320 and 2560 points, respectively). Also reported are Jacobi iterations

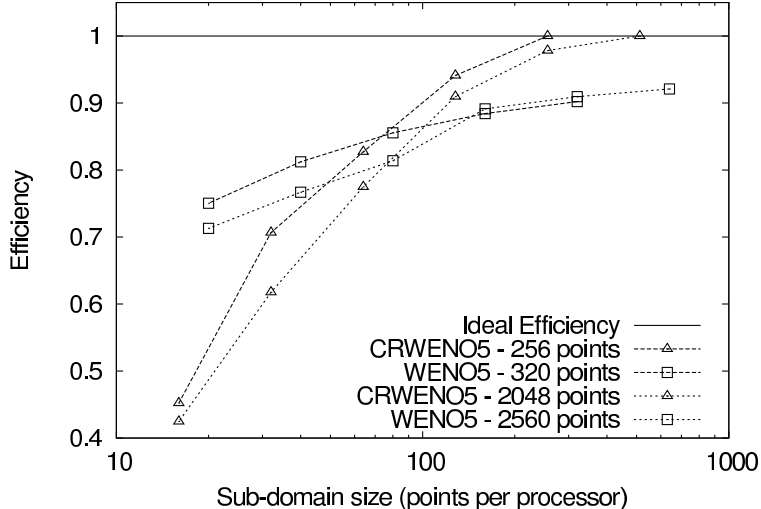


Figure 12: One-dimensional advection of density fluctuations. Efficiency is a function of the subdomain for the WENO5 and CRWENO5 schemes on grids with 256, 320, 2048, and 2560 points (data in Table 3).

(N_{Jac}) required by the CRWENO5 scheme to yield solutions identical to that obtained on a single processor. Once again, we observe that the number of Jacobi iterations increases as the number of processors increase and the subdomain size decreases. We note that the number of Jacobi iterations is not a function of the grid size but only of the subdomain size. We also observe that for both problem sizes, the CRWENO5 scheme is less expensive than the WENO5(F) scheme until a sub-domain size of 128 points; for lower subdomain sizes, the WENO5(F) scheme is less expensive. Figure 12 shows the efficiency of the CRWENO5 scheme (on grids with 256 and 2048 points) and the WENO5 scheme (on grids with 320 and 2560 points) as a function of the subdomain size. We observe that the CRWENO5 scheme is more efficient for the larger subdomain sizes; once again, it can be inferred that the critical subdomain size below which the CRWENO5 scheme is less efficient is insensitive to the global domain size.

The one-dimensional numerical experiments demonstrate the basic properties of our parallel implementation of the nonlinear compact schemes. The compact schemes yield solutions that are comparable to the solution obtained by the WENO5 scheme on a finer grid; however, on a single processor, they are less expensive than the WENO5 scheme on the finer grid. They are thus less expensive when considering solutions of the same accuracy and resolution. The cost of the tridiagonal solver increases as the number of processors increases for a given grid size because the number of Jacobi iterations needed to solve the reduced system to machine-zero accuracy increases. The relative cost of the CRWENO5 scheme increases as the subdomain size decreases; and below a critical subdomain size, the compact schemes are less efficient than the WENO5 scheme. We demonstrate that for subdomain sizes larger than this critical subdomain size, the CRWENO5 scheme is significantly more efficient. In addition, this critical subdomain size is independent of the global problem size, and thus this implementation is expected to scale for larger problems on correspondingly larger number of processors.

We now investigate the performance on two-dimensional problems. Solution to multidimensional problems with compact schemes require the solution to several tridiagonal systems of equations (one along each grid line along each dimension). We expect the efficiency of the tridiagonal solver to improve because of the increased arithmetic density of solving several systems together, along with more efficient communication with larger message sizes. In addition, the WENO5 scheme will need a larger number of grid points along each dimension thus amplifying the difference in the numerical cost of the CRWENO5 scheme on a coarser grid and the WENO5 on a finer mesh. Based on these factors, we expect the compact schemes to be more efficient than the WENO5 for smaller subdomain sizes per dimension. We start with a smooth problem with

Table 4: Errors (L_2) and wall times (in seconds) for the two-dimensional advection of a sinusoidal density wave.

N_{global}	N_{local}	WENO5		CRWENO5		
		Error	Wall Time	Error	Wall Time	N_{Jac}
4 (2^2) processors						
64^2	32^2	9.5348E-08	1.7501E+03	1.6491E-08	2.0155E+03	4
96^2	48^2	1.2567E-08	3.8659E+03	-	-	-
16 (4^2) processors						
64^2	16^2	9.5348E-08	4.5548E+02	1.6491E-08	5.5194E+02	4
96^2	24^2	1.2567E-08	9.9407E+02	-	-	-
128^2	32^2	2.9831E-09	1.7411E+03	3.2179E-10	2.0190E+03	4
192^2	48^2	3.9294E-10	3.8507E+03	-	-	-
64 (8^2) processors						
64^2	8^2	9.5348E-08	1.3187E+02	1.6491E-08	1.9005E+02	4
96^2	12^2	1.2567E-08	2.8053E+02	-	-	-
128^2	16^2	2.9831E-09	4.8237E+02	3.2179E-10	5.9309E+02	4
192^2	24^2	3.9294E-10	1.0530E+03	-	-	-
256^2	32^2	9.3256E-11	1.8441E+03	1.5378E-11	2.1623E+03	4
384^2	48^2	1.2283E-11	4.0651E+03	-	-	-
256 (16^2) processors						
64^2	8^2	9.5348E-08	3.9429E+01	1.6491E-08	1.0810E+02	8
96^2	12^2	1.2567E-08	7.8243E+01	-	-	-
128^2	8^2	2.9831E-09	1.3111E+02	3.2179E-10	2.0306E+02	6
192^2	12^2	3.9294E-10	2.7799E+02	-	-	-
256^2	16^2	9.3256E-11	4.7979E+02	1.5378E-11	5.9089E+02	4
384^2	24^2	1.2283E-11	1.0534E+03	-	-	-
1024 (32^2) processors						
128^2	8^2	2.9831E-09	3.9085E+01	3.2179E-10	1.1225E+02	10
192^2	12^2	3.9294E-10	7.8295E+01	-	-	-
256^2	8^2	9.3256E-11	1.3047E+02	1.5378E-11	2.0881E+02	8
384^2	12^2	1.2283E-11	2.9004E+02	-	-	-
4096 (64^2) processors						
256^2	8^2	9.3256E-11	3.8878E+01	1.5378E-11	1.2383E+02	12
384^2	12^2	1.2283E-11	7.7495E+01	-	-	-

the initial solution given by

$$\begin{aligned} \rho &= \rho_0 + \tilde{\rho} \sin(2\pi x) \cos(2\pi y), \\ u &= 1, \quad v = 1, \quad p = 1/\gamma, \end{aligned} \tag{30}$$

on a periodic two-dimensional domain $[0, 1] \times [0, 1]$. The mean density is $\rho_0 = 1$, the amplitude of the density wave is $\tilde{\rho} = 0.1$, and $\gamma = 1.4$ is the ratio of specific heats. Solutions are obtained with the WENO5 and CRWENO5 schemes on 64^2 , 128^2 , and 256^2 grids (baseline grids); in addition, solutions are obtained with the WENO5 scheme on grids that are 1.5 times as fine in each dimension (96^2 , 192^2 , and 384^2). This is a smooth problem, and the CRWENO5 scheme reduces to the linear fifth-order compact scheme. A small

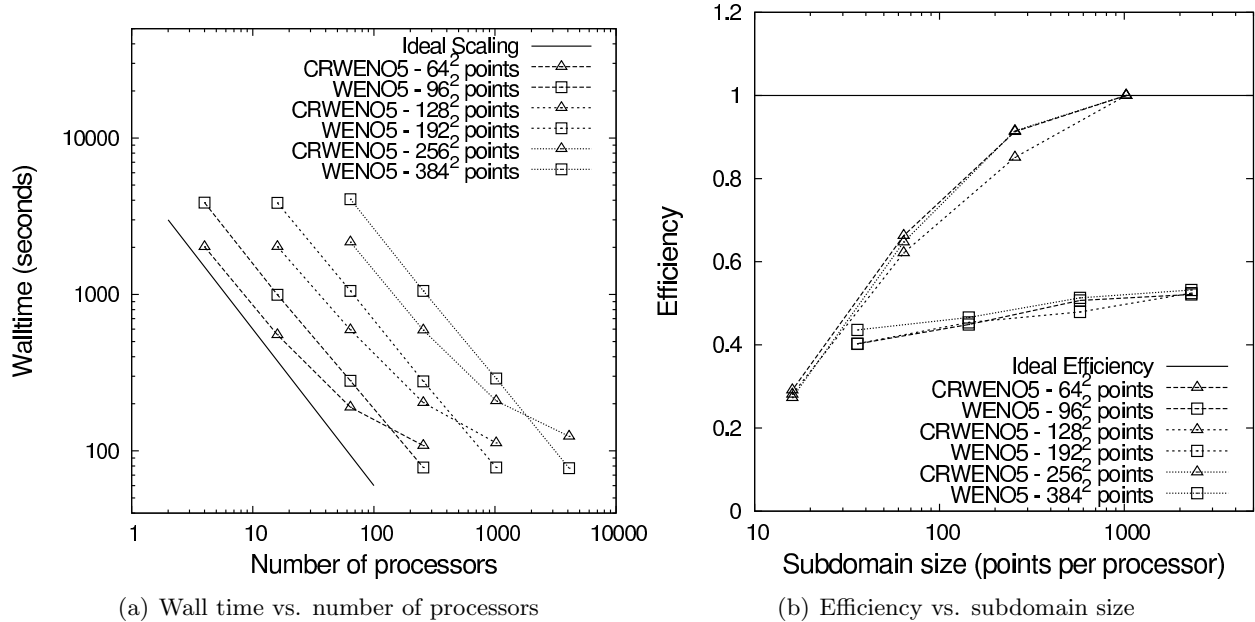


Figure 13: Two-dimensional advection of density sine wave – Wall times and efficiencies of the WENO5 and CRWENO5 schemes on grids with 64², 96², 128², 192², 256², and 384² points (data in Table 4).

time-step size of 10^{-4} is taken to keep time discretization errors to a minimum, and the solution is evolved for half a time period.

Table 4 shows the L_2 norm of the errors and wall times for the solutions obtained on the different grid sizes (N_{global}) and subdomain sizes (N_{local}). It is verified that both the schemes show fifth-order convergence for this two-dimensional smooth problem. The CRWENO5 scheme yields solutions with errors that are an order of magnitude lower than that of the WENO5 scheme on the same grid. The accuracy of the CRWENO5 scheme is comparable to the WENO5 scheme on grids that are 1.5 times as fine in each dimension. These observations are consistent with the Taylor series analysis of these schemes. The difference in the computational efficiencies of the WENO5 and CRWENO5 is starker in two dimensions than in one dimension. It is also verified that the errors in the solutions for a given grid size are identical for all the subdomain sizes, thus ensuring that the parallel implementation introduces no errors. The number of Jacobi iterations needed to ensure this increases as the subdomain size decreases.

We compare the wall times for the CRWENO5 scheme on the baseline grids with the WENO5 scheme on the 1.5²x finer grids. Similar to the one-dimensional results, the CRWENO5 scheme is less expensive than the WENO5 scheme for larger subdomain sizes. As the subdomain size decreases, the number of Jacobi iterations and consequently the cost of the CRWENO5 scheme increase. Thus, below a critical subdomain size, the CRWENO5 scheme is more expensive than the WENO5 scheme. For example, for the CRWENO5 scheme on the 64²-points grid and the WENO5 scheme on the 96²-points grid, the CRWENO5 scheme is more efficient for 4, 16, and 64 processors (subdomain sizes of 32², 16², and 8²) but less efficient for 256 processors (subdomain size of 4²). Similarly, for the CRWENO5 scheme on the 256²-points grid and the WENO5 scheme on the 384²-points grid, the CRWENO5 scheme is less expensive on 64, 256, and 1024 processors (subdomain sizes of 32², 16², and 8²) but more expensive on 4096 processors (sub-domain size of 4²). These observations are reiterated through Figure 13(a), where the wall times are shown as a function of the number of processors for the CRWENO5 and WENO5 schemes of comparable accuracy. We observe that for all the grid sizes considered, CRWENO5 is more efficient until a subdomain size of 8² points per processor; it is less efficient for smaller subdomains. Figure 13(b) shows the efficiency of the CRWENO5 and the WENO5 schemes as

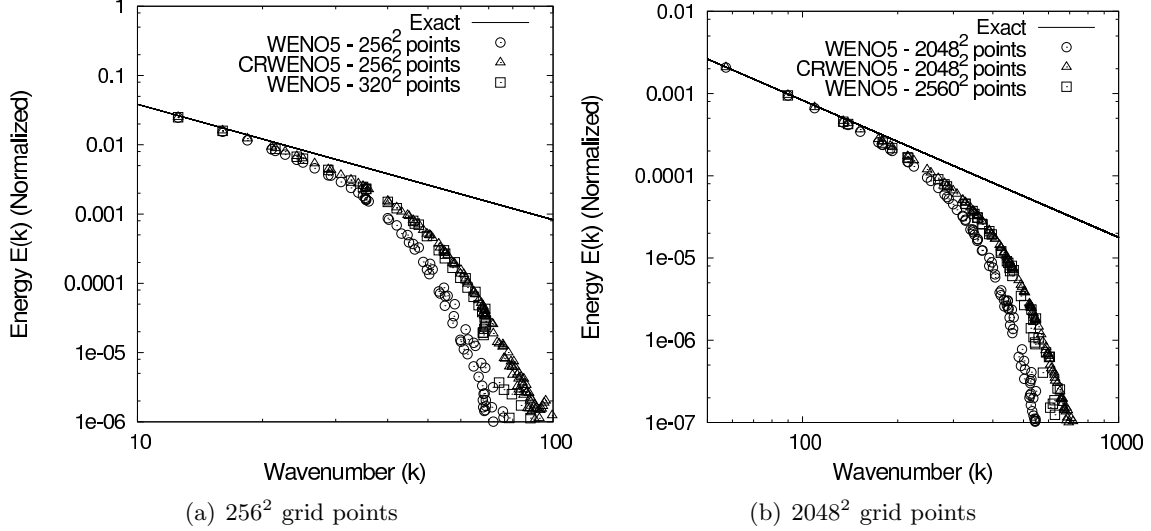


Figure 14: Energy spectrum of the numerical solutions to the two-dimensional advection of density fluctuations.

a function of the sub-domain size. CRWENO5 is observed to be less efficient than WENO5 scheme for subdomain sizes smaller than a critical value, and this critical subdomain size is insensitive to the global grid sizes (as was observed for the one-dimensional problems as well). We also observe that in two dimensions, the CRWENO5 scheme is more efficient until a smaller subdomain size per dimension, namely, 8 points per processor per dimension, as opposed to 64 points per processor for one dimension.

The performance of the schemes is assessed for the two-dimensional advection of density fluctuations with a prescribed energy spectrum. The density fluctuations in the initial solution are prescribed in the Fourier space as follows:

$$\hat{\rho}(k_x, k_y) = \frac{\tilde{\rho} |\mathbf{k}|^{-5/6}}{\sqrt{2}} (1 + i); \quad |\mathbf{k}| = \sqrt{k_x^2 + k_y^2}; \quad 1 \leq k_x, k_y \leq N/2, \quad (31)$$

where N is the number of points per dimension on a square grid (the complex conjugates are taken in the remainder of the domain $k_x, k_y > N/2$ to ensure that the fluctuations in the physical space are real). The initial density, velocity, and pressure are then prescribed as

$$\begin{aligned} \rho &= \rho_0 + \delta\rho, \\ u &= 1, \quad v = 1, \quad p = 1/\gamma, \end{aligned} \quad (32)$$

where $\delta\rho(x, y)$ is the inverse Fourier transform of $\hat{\rho}(k_x, k_y)$. The maximum amplitude of fluctuations $\tilde{\rho}$ is taken as 10^{-2} to ensure that the total density is non-negative. The two-dimensional domain is $[0, 1] \times [0, 1]$ and periodicity is enforced at all boundaries.

The problem is studied at two baseline grid sizes, 256^2 and 2048^2 points, to show that the critical subdomain size is independent of the problem size. Solutions are obtained at $t = 0.1T$, where T is the time period with time-step sizes of 2×10^{-4} and 2×10^{-5} , respectively for the two baseline grids. Figure 14 shows the energy spectrum of the solutions obtained by the WENO5 and CRWENO5 schemes on the 256^2 and 2048^2 grids. Also shown is the spectrum of the solutions obtained by the WENO5 scheme on grids with 320^2 and 2560^2 points. One can see that the spectral resolution of the CRWENO5 scheme on the baseline grids is comparable to that of the WENO5 scheme on the grids with 1.25² times more points. Table 5 shows the wall times for the WENO5 and CRWENO5 schemes on the baseline grids (256^2 and 2048^2) as well as the

Table 5: Wall times (in seconds) for the two-dimensional advection of density fluctuations.

N_{proc}	N_{local}	Wall times			N_{Jac}
		WENO5	WENO5(F)	CRWENO5	
256 ² grid points					
4	128 ² (160 ²)	2.7048E+03	4.2190E+03	3.0583E+03	0
16	64 ² (80 ²)	6.9221E+02	1.0787E+03	7.7876E+02	0
64	32 ² (40 ²)	1.8919E+02	2.9329E+02	2.1905E+02	2
256	16 ² (20 ²)	4.9332E+01	7.6092E+01	6.1110E+01	4
1024	8 ² (10 ²)	1.3410E+01	2.2011E+01	2.0199E+01	6
4096	4 ² (5 ²)	4.1890E+00	5.9152E+00	9.3200E+00	6
2048 ² grid points					
256	128 ² (160 ²)	2.9113E+03	4.5533E+03	3.3723E+03	0
1024	64 ² (80 ²)	7.4115E+02	1.1608E+03	8.6077E+02	2
4096	32 ² (40 ²)	1.8899E+02	2.9354E+02	2.2130E+02	4
16384	16 ² (20 ²)	5.2870E+01	8.3077E+01	6.0775E+01	6

Number in parentheses is the N_{local} for the WENO5 scheme on the finer grid (“WENO5(F)”)

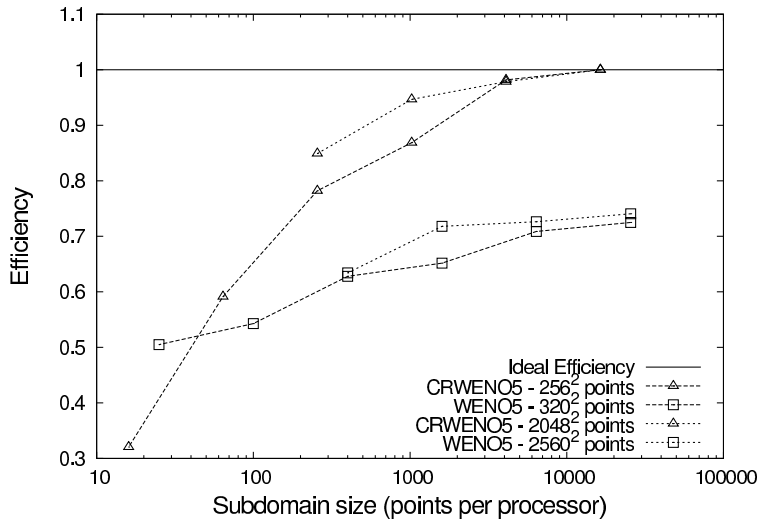


Figure 15: Two-dimensional advection of density fluctuations – Efficiency as a function of the subdomain for the WENO5 and CRWENO5 schemes on grids with 256², 320², 2048², and 2560² points (data in Table 5).

WENO5 scheme on the 320² and 2560² points in column “WENO5(F).” The number of processors (N_{proc}) and the corresponding subdomain sizes (N_{local}) are reported in the first two columns, and the number of Jacobi iterations (N_{Jac}) required by the CRWENO5 scheme is reported in the last column. We observe that for the baseline grid of 256² points, the CRWENO5 and HCWENO5 schemes are less expensive than the WENO5 scheme until a subdomain size of 8 points per dimension per processor. The WENO5 scheme is more efficient for subdomain sizes lower than this because of the increasing cost of the tridiagonal solver. Because of hardware restrictions, the case with the baseline grid of 2048² points could not be studied at subdomain sizes smaller than 16² points. We observe that the CRWENO5 scheme is significantly more efficient than the WENO5 scheme for all the subdomain sizes considered for this case. Figure 15 shows the efficiency as

Table 6: Errors (L_2) and wall times (in seconds) for the three-dimensional advection of a sinusoidal density wave.

N_{global}	N_{local}	WENO5		CRWENO5		
		Error	Wall Time	Error	Wall Time	N_{Jac}
64 (4^3) processors						
32^3	8^3	4.0388E-06	4.1216E+03	5.4544E-07	5.0583E+03	10
48^3	12^3	5.3211E-07	1.2960E+04	-	-	-
512 (8^3) processors						
32^3	4^3	4.0388E-06	6.0777E+02	5.4544E-07	9.3257E+02	10
48^3	6^3	5.3211E-07	1.8179E+03	-	-	-
64^3	8^3	1.2639E-07	4.0552E+03	1.4849E-08	4.9851E+03	10
96^3	12^3	1.6660E-08	1.2884E+04	-	-	-
4096 (16^3) processors						
64^3	4^3	1.2639E-07	6.0635E+02	1.4849E-08	9.3614E+02	10
96^3	6^3	1.6660E-08	1.8033E+03	-	-	-
128^3	8^3	3.9549E-09	4.0209E+03	4.3038E-10	4.9291E+03	10
192^3	12^3	5.2096E-10	1.2819E+04	-	-	-
32768 (32^3) processors						
128^3	4^3	3.9549E-09	6.5648E+02	4.3038E-10	9.4118E+02	10
192^3	6^3	5.2096E-10	1.9532E+03	-	-	-

a function of the subdomain size for the CRWENO5 and WENO5 schemes, and the previous observations are reiterated. Although the wall times and efficiencies are not available for smaller subdomain sizes on the 2048^2 and 2560^2 point grids, the efficiencies for the CRWENO5 and WENO5 schemes show a trend similar to that observed on the 256^2 and 320^2 point grids.

We also investigated the performance of the schemes for three-dimensional problems. Similar to the two-dimensional problems, the tridiagonal solver is expected to be highly efficient for three-dimensional problems by solving several systems of equations together (thus maximizing communication efficiency). In addition, the difference in the computational efficiencies for the WENO5 and CRWENO5 schemes is further amplified by the dimensionality—the WENO5 scheme requires a grid with 1.5^3 times more points than the compact schemes for smooth problems. Thus, the critical subdomain size for the compact schemes is expected to be smaller per dimension for three-dimensional problems. We consider a smooth problem with the initial solution specified as follows:

$$\begin{aligned} \rho &= \rho_0 + \tilde{\rho} \sin(2\pi x) \sin(2\pi y) \sin(2\pi z), \\ u &= 1, \quad v = 1, \quad w = 1, \quad p = 1/\gamma, \end{aligned} \tag{33}$$

where the mean density is $\rho_0 = 1$ and the amplitude of the wave is $\tilde{\rho} = 0.1$. The specific heat ration is $\gamma = 1.4$. Solutions are obtained with the CRWENO5 and WENO5 schemes on 32^3 , 64^3 , and 128^3 grids (baseline); in addition, solutions are obtained with the WENO5 scheme on 48^3 , 96^3 , and 192^3 grids. A time-step size of 5×10^{-5} is taken to minimize errors due to time discretization, and solutions are obtained at half a time period.

Table 6 reports the errors (L_2 norm) and wall times for the grid sizes (N_{global}) and subdomain sizes (N_{local}) considered. The choice of cases is constrained by hardware and job scheduling limits on the computing platform. Both the WENO5 and CRWENO5 schemes show fifth-order convergence as expected, and the errors in the solutions obtained by the CRWENO5 scheme are an order of magnitude lower than those in the solutions obtained by the WENO5 scheme. We compare the numerical cost of the CRWENO5 scheme

on the baseline grids (32^3 , 64^3 , and 128^3 points) with that of the WENO5 scheme on the grids with 1.5^3 times the number of points (48^3 , 96^3 , and 192^3) since these cases yield solutions of comparable accuracy. It is observed that for all the cases considered, the CRWENO5 scheme is less expensive than the WENO5 scheme for solutions of comparable accuracy, including the smallest subdomain size of 4^3 points per processor. The number of Jacobi iterations (N_{Jac} in Table 6 is identical (10) for all the cases reported. Although ten subiterations is more than the minimum number of Jacobi iterations required for machine-zero convergence, especially for the cases with a subdomain size of 8^3 points, the CRWENO5 scheme is less expensive by a relatively large margin, and this allows us to specify a more-than-adequate number of iterations. Thus, all the cases reported carried out 10 Jacobi iterations to solve the reduced tridiagonal system.

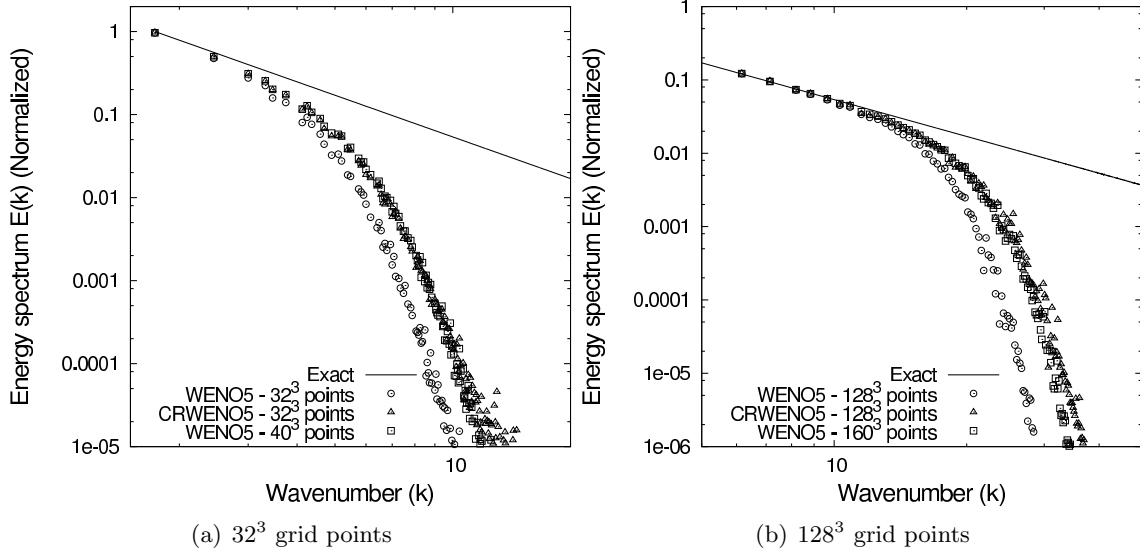


Figure 16: Energy spectrum of the numerical solutions to the three-dimensional advection of density fluctuations.

Table 7: Wall times (in seconds) for the three-dimensional advection of density fluctuations.

N_{proc}	N_{local}	Wall times			N_{Jac} CRWENO5
		WENO5	WENO5(F)	CRWENO5	
32 ³ grid points					
8	16 ³ (20 ³)	2.9409E+03	5.6461E+03	3.3439E+03	8
64	8 ³ (10 ³)	4.3216E+02	8.1914E+02	5.2010E+02	8
512	4 ³ (5 ³)	6.4237E+01	1.1666E+02	9.2764E+01	8
128 ³ grid points					
8	64 ³ (80 ³)	6.4556E+05	1.2562E+06	7.9542E+05	8
64	32 ³ (40 ³)	8.9239E+04	1.7233E+05	1.0331E+05	8
512	16 ³ (20 ³)	1.2272E+04	2.3489E+04	1.4651E+04	8
4096	8 ³ (10 ³)	1.7224E+03	3.2696E+03	2.0835E+03	8
32768	4 ³ (5 ³)	2.5811E+02	4.7065E+02	3.7761E+02	8

Number in parentheses is the N_{local} for the WENO5 scheme on the finer grid (“WENO5(F)”)

The performance of the schemes for the periodic advection of density fluctuations with a specified energy spectrum is studied by prescribing the initial fluctuations in the Fourier space:

$$\hat{\rho}(k_x, k_y, k_z) = \frac{\tilde{\rho} |\mathbf{k}|^{-5/6}}{\sqrt{2}} (1 + i); \quad |\mathbf{k}| = \sqrt{k_x^2 + k_y^2 + k_z^2}; \quad 1 \leq k_x, k_y, k_z \leq N/2, \quad (34)$$

where N is the number of points per dimension on a square grid and the complex conjugates are taken in the remainder of the wavenumber domain. The initial density, velocity, and pressure are

$$\begin{aligned} \rho &= \rho_0 + \delta\rho, \\ u &= v = w = 1, \quad p = 1/\gamma, \end{aligned} \quad (35)$$

where $\delta\rho(x, y, z)$ is the inverse Fourier transform of $\hat{\rho}(k_x, k_y, k_z)$. The maximum amplitude of fluctuations $\tilde{\rho}$ is taken as 10^{-5} to ensure that the total density is non-negative. A periodic three-dimensional domain $([0, 1]^3)$ is taken.

Solutions are obtained for two baseline grids— 32^3 and 128^3 points—with the WENO5 and CRWENO5 schemes. Time-step sizes of 10^{-3} and 2.5×10^{-4} are taken for the two grids, respectively, and the solution is evolved for one time period. In addition, corresponding to the two baseline grids, solutions are obtained on grids with 40^3 and 160^3 points with the WENO5 scheme with the same time step sizes. Figure 16 shows the energy spectrum of the numerical solutions obtained with the CRWENO5 and WENO5 schemes. One can see that for both the baseline grids the CRWENO5 scheme exhibits a spectral resolution on the baseline grids (32^3 and 128^3 points) that is comparable to that of the WENO5 scheme on grids with 1.25^3 more points (40^3 and 160^3). Table 7 reports the wall times for the cases (N_{proc} is the number of processors and N_{local} is the corresponding subdomain size). The CRWENO5 scheme is significantly more efficient than the WENO5 scheme for all the cases considered. The reported number of Jacobi iterations (N_{Jac}) is identical (8) for all the cases and is more than adequate to ensure machine-zero convergence. The performance margin between the WENO5 scheme and the compact schemes is relatively large; and despite a large (possibly redundant) number of Jacobi iterations, the compact schemes are less expensive. Figure 17(a) shows the wall times as a function of the number of processors for the WENO5 and CRWENO5 schemes. The CRWENO5 scheme is less expensive than the WENO5 scheme for all the cases at both the grid sizes considered. Figure 17(b) shows the efficiency as a function of the subdomain sizes. One can see that the CRWENO5 scheme is more efficient than the WENO5 scheme even at the smallest subdomain size of 64 (4^3) points per processor (5^3 points for WENO5). It should be noted that fifth-order schemes such as CRWENO5 and WENO5 require three ghost points along each dimension for data exchange between neighboring subdomains, and thus subdomains smaller than 4^3 points are not practical.

The numerical experiments presented in this section demonstrate that our implementation of the parallel tridiagonal solver does not introduce any parallelization-related errors if the reduced system is solved to machine-zero convergence. The diagonal dominance of the reduced system ensures that this is possible within a reasonable number of iterations. Unlike previous approaches [13, 38], our method yields solutions on multiprocessors that are identical to those obtained on a single processor. The mathematical complexity of the tridiagonal solver is comparable to that of the Thomas algorithm on the single processor. In addition, our algorithm avoids collective communications as well as the huge communication requirements of the transposition approach [16, 29]. A larger number of processors for a given problem size (consequently smaller subdomains) result in reduced diagonal dominance of the reduced system and a larger number of Jacobi iterations needed to ensure convergence. The relative cost of the tridiagonal solver thus increases with the number of processors. CRWENO5 is more efficient than WENO5 for subdomain sizes larger than a critical size, below which the cost of the tridiagonal solver renders them more expensive. This is shown for one and two spatial dimensions. This critical subdomain size is observed to be independent of the global problem size, and thus this algorithm can be scaled up for larger problems on a correspondingly larger number of processors.

Numerical experiments in three space dimensions show that the CRWENO5 scheme is more computationally efficient for the smallest practical subdomain sizes possible— 4^3 points per processor. One can see from

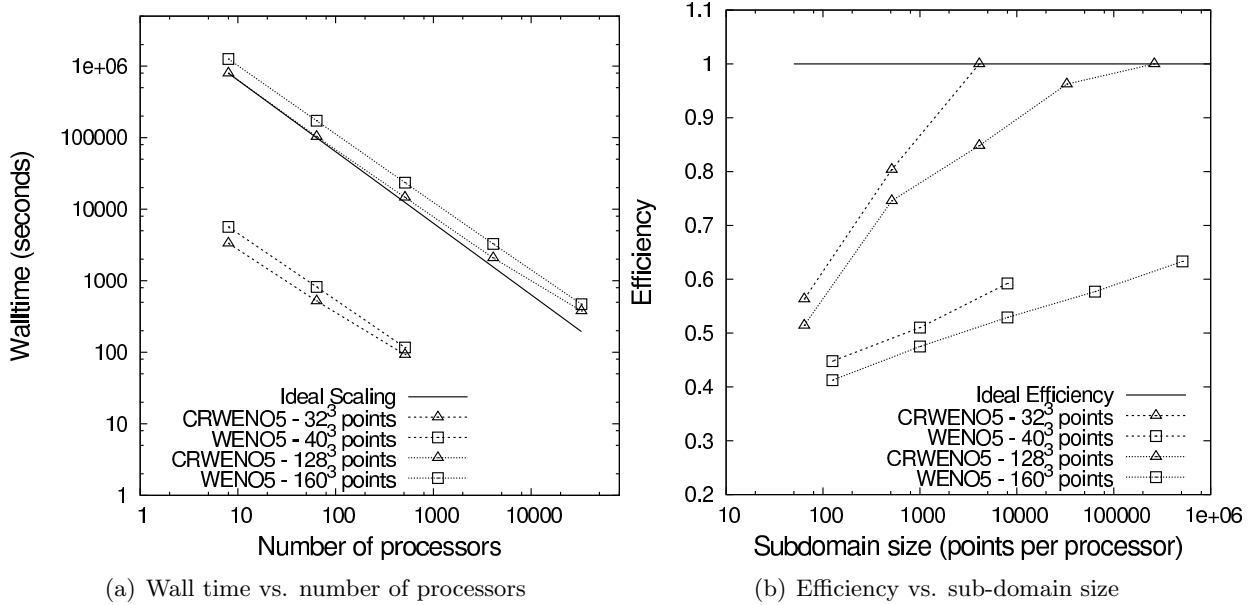


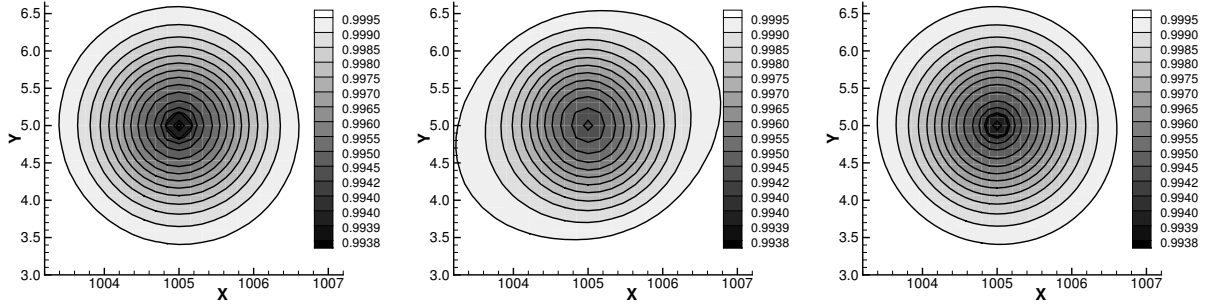
Figure 17: Three-dimensional advection of density fluctuations – Wall times and efficiencies of the WENO5 and CRWENO5 schemes on grids with 32^3 , 40^3 , 128^3 , and 160^3 points (data in Table 7).

the wall times and efficiencies that, although the parallel implementation of the CRWENO5 scheme does not scale as well as the noncompact WENO5 scheme, the absolute numerical cost is still lower than that of the WENO5 scheme on a finer grid that yields solutions of comparable accuracy and resolution. That is, for a given number of processor, it is less expensive to use the CRWENO5 scheme to obtain a solution of the desired accuracy or resolution. We can thus claim that our implementation of the compact schemes for three-dimensional problems is more efficient for the entire range of parallelization, including massively parallel fine-grained simulations.

4 Results

The previous section analyzed the numerical performance of the parallel CRWENO5 scheme through manufactured solutions involving the advection of density waves. The performance of the overall algorithm is evaluated in this section on benchmark flow problems representative of practical flows. Previous studies on the CRWENO5 scheme [25–27] demonstrated two desirable benefits compared with the standard WENO5 scheme apart from sharper resolution of discontinuities: accurate preservation of flow features as they convect large distances and improved resolution of a larger range of relevant length scales for turbulent flows. The two flow problems in this section—the long-term convection of an isentropic vortex and the decay of isotropic turbulence—illustrate these properties. Moreover, the computational efficiency of the CRWENO5 scheme on multiple processors is demonstrated for these flows.

The computations presented in this section are carried out on Mira, a Blue Gene/Q system maintained by the ALCF [1]. The system has 491,52 compute nodes, each with a 16-core 1600 MHz PowerPC A2 processor. Each core can support 4 hardware threads. Each node has a 16 GB RAM and the overall system has a peak performance of 10 petaflops. Our choice of test cases and problem sizes attempts maximum possible utilization of these resources within the constraints of our project allocation.



(a) CRWENO5 - $8192 \times 64 \times 64$ grid (b) WENO5 - $8192 \times 64 \times 64$ grid (c) WENO5 - $12288 \times 96 \times 96$ grid

Figure 18: Isentropic vortex convection: density contours after vortex has traveled a distance 1,000 times its core radius.

4.1 Isentropic Vortex Convection

The isentropic vortex is an exact solution of the inviscid Euler equations [58] and convects with the freestream flow with its shape and strength preserved. The numerical simulation of the long-term convection of the isentropic vortex tests the ability of the algorithm to preserve a flow feature for large simulation times. The vortex is a two-dimensional flow; however, we solve this flow over a three-dimensional domain to demonstrate the computational cost and efficiency of the three-dimensional solver. Periodic boundary conditions are imposed along the third (normal) dimension. It was demonstrated previously [27] that the CRWENO5 scheme shows a significant improvement in the preservation of the strength and shape of the vortex as it convects over a large distance (1000 times the core radius). These results were obtained on a small, two-dimensional domain (with edge length as 10 times the core radius) with periodic boundary conditions ensuring that the vortex traveled the length of this domain for several cycles.

In this study, we consider a large domain along the direction of vortex convection, in order to evaluate the strong and weak scaling of the parallel algorithm for a large number of grid points and correspondingly large number of processors. The freestream flow is specified as $\rho_\infty = 1$ (density), $u_\infty = 0.5$ (x -velocity), $v_\infty = w_\infty = 0$ (y and z velocities), and $p_\infty = 1$ (pressure). The initial vortex is specified as

$$\begin{aligned}
 \rho &= \left[1 - \frac{(\gamma - 1)b^2}{8\gamma\pi^2} e^{1-r^2} \right]^{\frac{1}{\gamma-1}}, \\
 \delta u &= -\frac{b}{2\pi} e^{\frac{1-r^2}{2}} (y - y_c), \\
 \delta v &= \frac{b}{2\pi} e^{\frac{1-r^2}{2}} (x - x_c), \\
 \delta w &= 0, \quad p = \rho^\gamma,
 \end{aligned} \tag{36}$$

where δu , δv , and δw are the velocity perturbations, $(x_c, y_c) = (5, 5)$ is the initial location of the vortex center, $r = (x^2 + y^2)^{1/2}$ is the radial distance from the vortex center, and $b = 0.5$ is the vortex strength. The flow is uniform along the z dimension. Periodic boundary conditions are specified on all boundaries.

The strong scaling of the algorithm is evaluated by solving the flow on a domain of length $1280R \times 10R \times 10R$ (where $R = 1$ is the vortex core radius), discretized by a grid with $8192 \times 64 \times 64$ points. Solutions are obtained with the CRWENO5 and WENO5 schemes. In addition, solutions are obtained with the WENO5 scheme on a grid with $12288 \times 96 \times 96$ points. The vortex is allowed to convect a distance of 1000 times the core radius. The solution is integrated in time by using the third-order accurate strong-stability-preserving

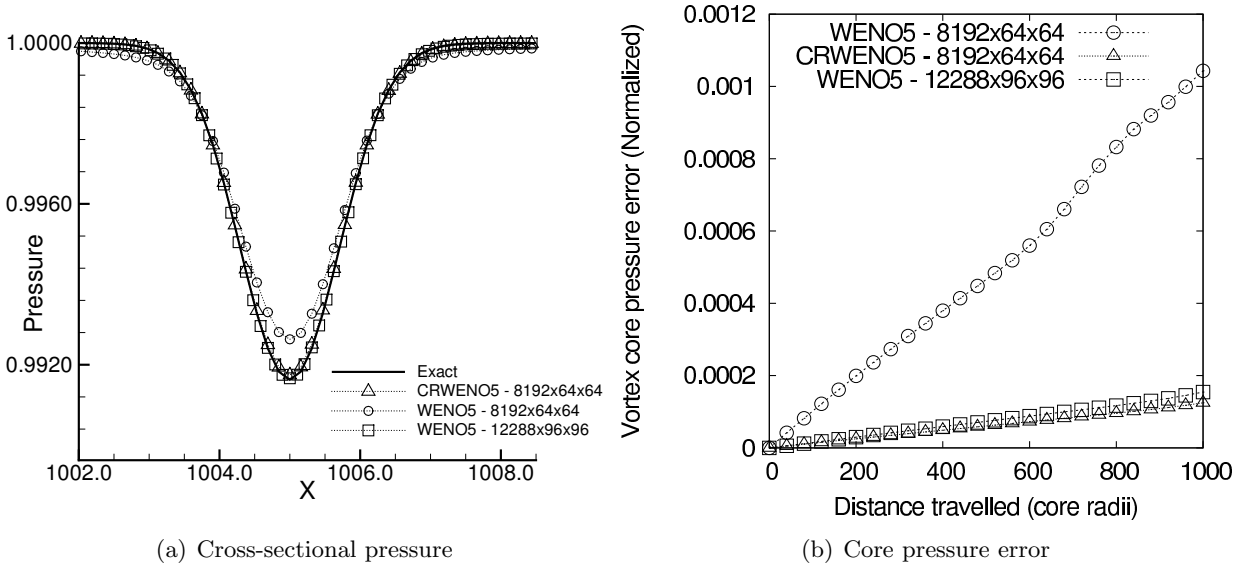


Figure 19: Isentropic vortex convection: cross-sectional pressure after vortex has traveled a distance 1,000 times its core radius, and pressure error at vortex center as a function of time.

Runge-Kutta (SSPRK3) scheme [32] with a time-step size of 0.025 units. Figure 18 shows the density contours of the flow for the solutions obtained with the WENO5 and CRWENO5 schemes. One can see that the solution obtained by the WENO5 scheme on the $8192 \times 64 \times 64$ point grid is dissipated, while the CRWENO5 scheme yields a solution comparable to that obtained by the WENO5 scheme on the $12288 \times 96 \times 96$ point grid. This is reiterated in Figure 19(a), which shows the cross-sectional pressure through the vortex core. Figure 19(b) shows the error in the vortex core pressure (normalized by the initial core pressure) as a function of the distance traveled. These figures demonstrate that the CRWENO5 scheme yields solutions that are comparable to that of the WENO5 scheme on a grid with 1.5^3 times more points.

We thus compare the wall times of the CRWENO5 scheme on the $8192 \times 64 \times 64$ points grid with those of the WENO5 scheme on the $12288 \times 96 \times 96$ points grid. The number of Jacobi iterations for the CRWENO5 scheme is fixed at 10 irrespective of the subdomain size. The domain is partitioned along all three dimensions. Figure 20(a) shows the wall times (for one Runge-Kutta time-integration stage) as a function of the number of processors. The subdomain sizes range from 4^3 (6^3 for WENO5) for 524288 ($2048 \times 16 \times 16$) processors to 16^3 (24^3 for WENO5) for 8192 ($512 \times 4 \times 4$) processors. We observe that although the CRWENO5 scheme does not scale as well as the WENO5 scheme, the absolute wall time is significantly lower. Figure 20(b) compares the efficiencies of the two schemes as a function of the subdomain size. The efficiency of the CRWENO5 scheme decreases rapidly as the subdomain size decreases; however, in absolute terms, the CRWENO5 is significantly more efficient than the WENO5 scheme even for the smallest subdomain size.

Figure 20(c) shows the Runge-Kutta stage wall times of the CRWENO5 and WENO5 scheme for constant subdomain sizes of 4^3 and 6^3 points, respectively, as the number of grid points and the number of processors are increased by the same factors. These results are obtained by varying the physical length, number of points, and number of processors along the direction of vortex convection while keeping these quantities along the other two dimensions constant. We initially start with a domain of size $40R \times 10R \times 10R$, discretized by a grid with $256 \times 64 \times 64$ points ($384 \times 96 \times 96$ points for WENO5) on 16,384 ($64 \times 16 \times 16$) processors and increase the quantities in the x -dimension by a factor of two till a domain of size $1280R \times 10R \times 10R$, discretized by a grid with $8192 \times 64 \times 64$ points ($12288 \times 96 \times 96$ points for WENO5) on 524,288 ($2048 \times 16 \times 16$) processors. The wall times for the CRWENO5 scheme are significantly lower than those of the WENO5

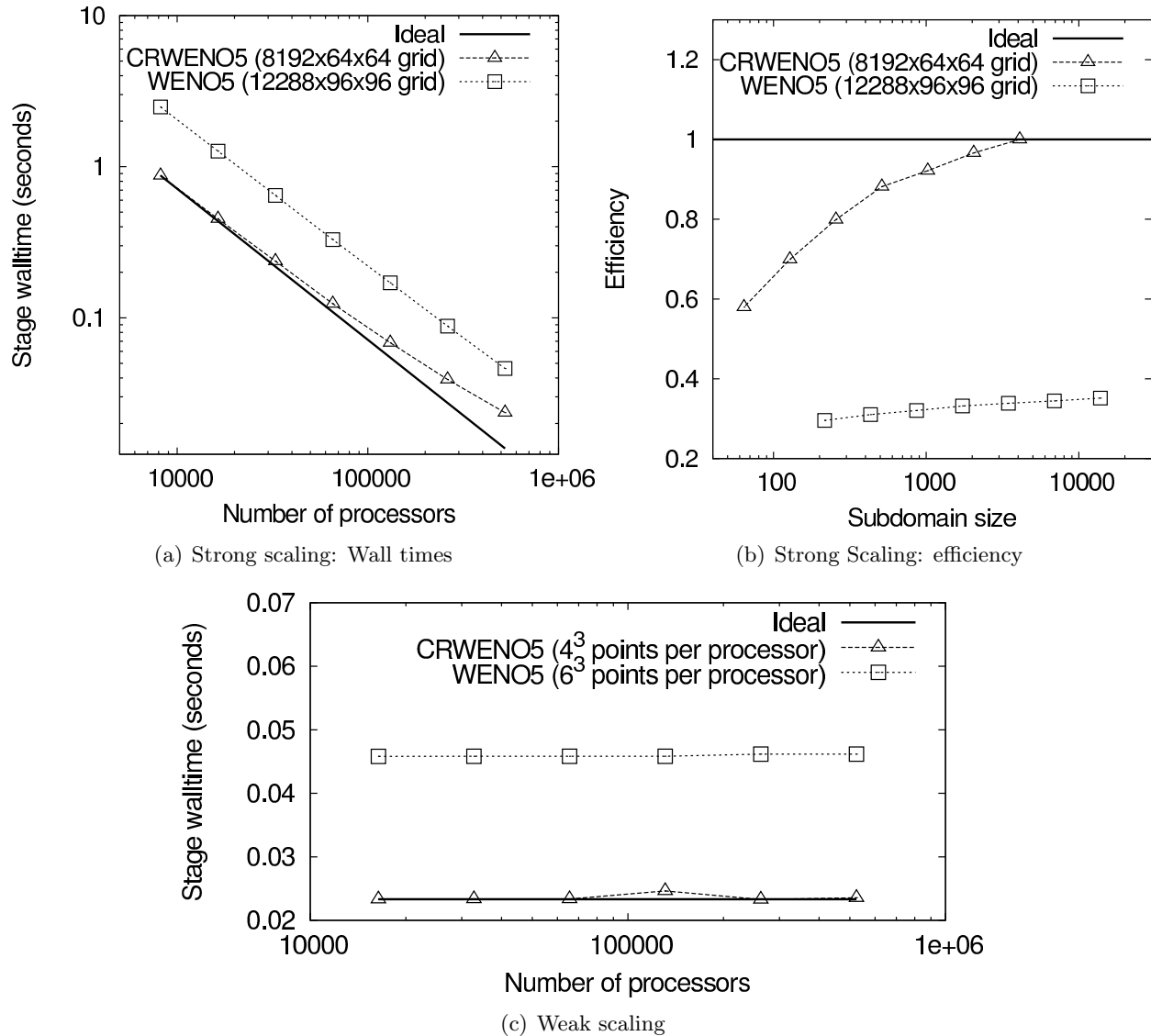


Figure 20: Isentropic vortex convection: wall times and parallel efficiencies for the CRWENO5 and WENO5 schemes.

scheme. The parallel implementation of the tridiagonal solver involves only point-to-point communications between processors and thus, an excellent weak scaling is observed. We can conclude that the CRWENO5 scheme will remain more efficient than the WENO5 scheme as the problem size and the number of processors increase further.

4.2 Isotropic Turbulence Decay

The decay of an isotropic turbulent flowfield [46, 55] is a benchmark problem characterized by a transfer of energy from larger to smaller length scales. An initial solenoidal velocity field is specified that satisfies a prescribed kinetic energy spectrum, and this problem tests the ability of the algorithm to model moderate and small length scales accurately. In addition, the flow is compressible for higher values of the turbulent

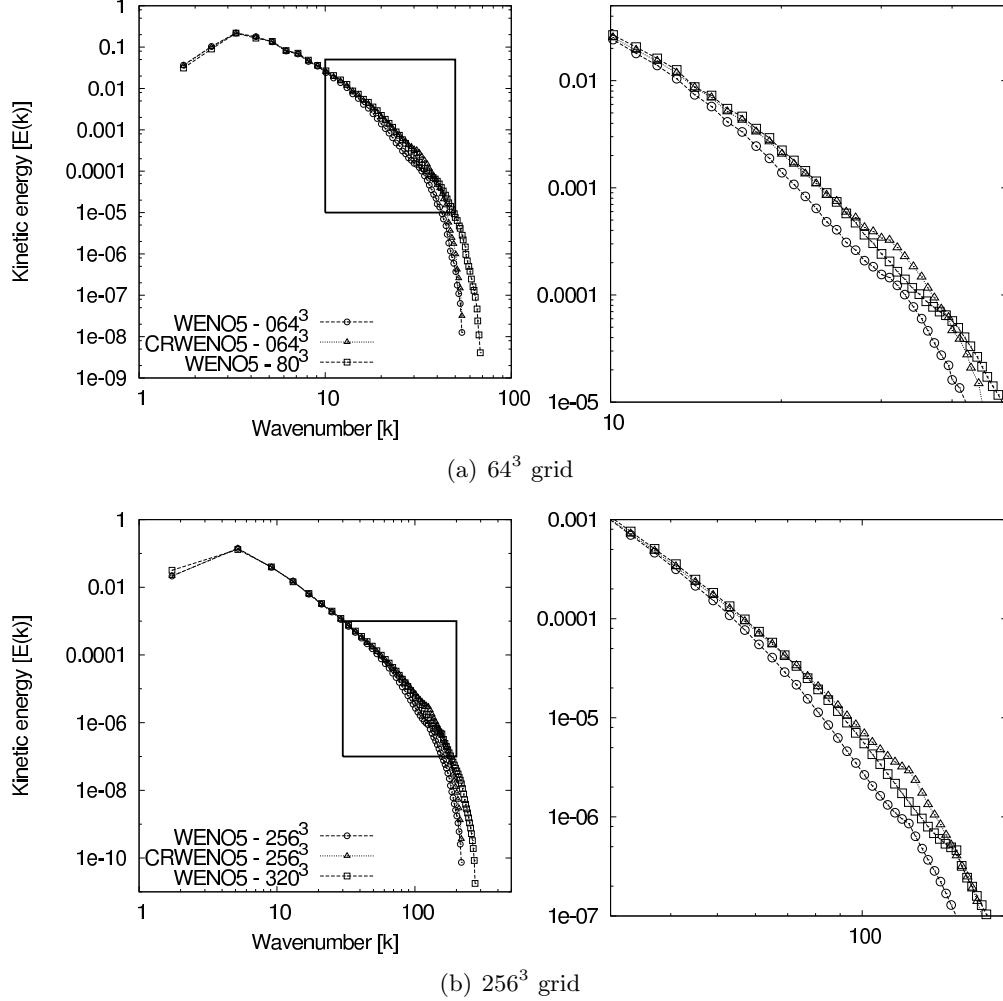


Figure 21: Isotropic turbulence decay: Energy spectrum at $t = 3.0$ for solutions obtained by the WENO5 and CRWENO5 schemes.

fluctuations, and a non-oscillatory scheme is required. Previous studies [25, 26] have demonstrated through direct numerical simulation that the CRWENO5 scheme yields solutions with higher resolution of moderate and high wavenumbers. In the present study, an implicit large-eddy simulation [65] is presented where the inviscid Euler equations are solved and the numerical viscosity provides the sub-grid model.

The physical domain is a periodic cube with edge length 2π units, and the initial velocity field is specified in the Fourier space such that its divergence is zero and the kinetic energy spectrum satisfies [25, 41]:

$$E(k) = 16\sqrt{\frac{2}{\pi}} \frac{u_0^2}{k_0} \left(\frac{k}{k_0}\right)^4 \exp\left[-2\left(\frac{k}{k_0}\right)^2\right], \quad (37)$$

where E is the kinetic energy, k is the wavenumber, $k_0 = 4$ is the wavenumber corresponding to the maximum kinetic energy, and $u_0 = 0.2\sqrt{2}$ is the RMS turbulence intensity. Constant initial density and pressure are specified ($\rho = 1$ and $p = 1/\gamma$). The problem is solved with the WENO5 and CRWENO5 schemes on two grids— 64^3 and 256^3 points—and with the WENO5 scheme on grids that are 1.25^3 times as fine (80^3 and 320^3 points). The number of Jacobi iterations for the tridiagonal solver in the CRWENO5 scheme is fixed at

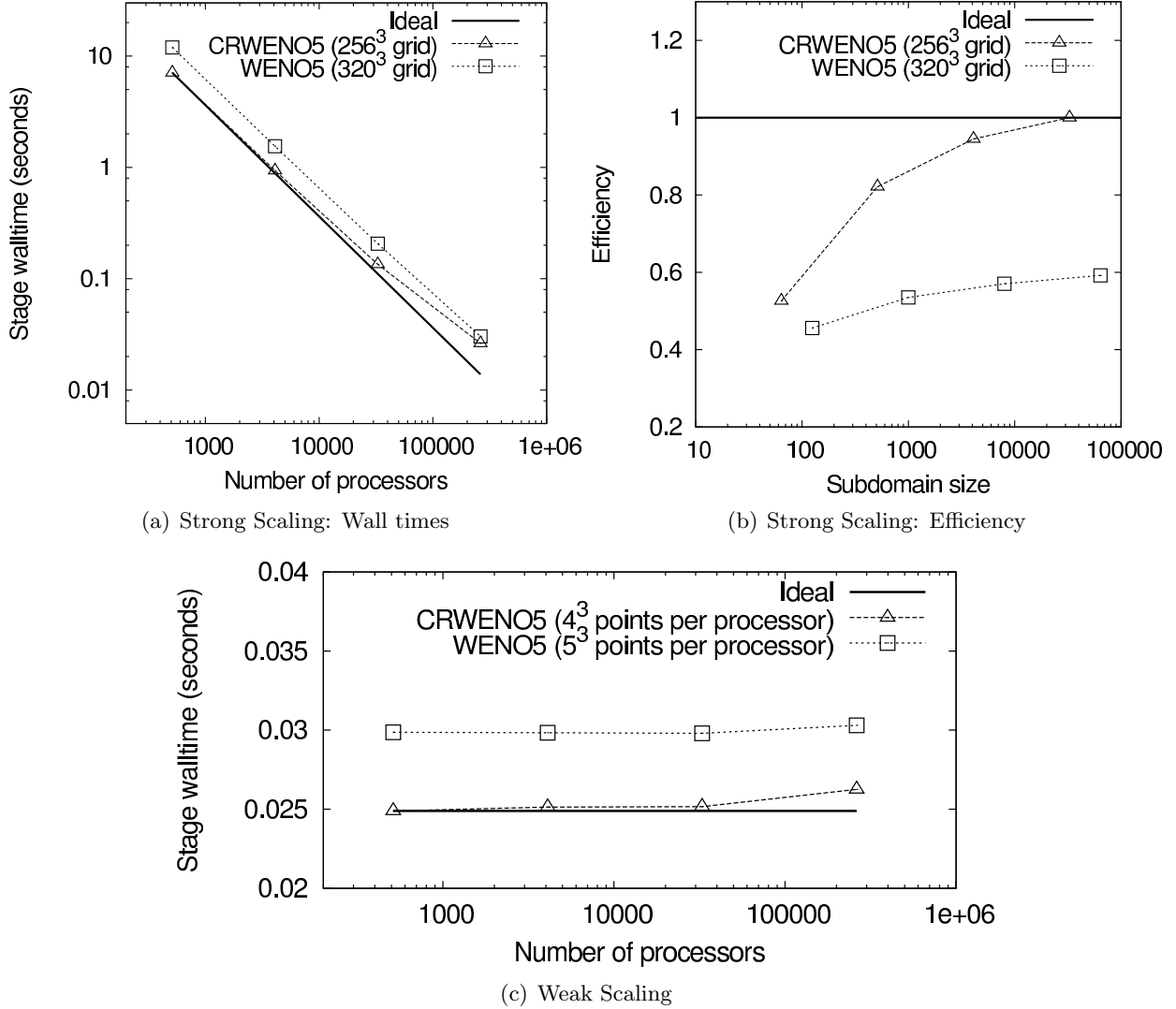


Figure 22: Isotropic turbulence decay: wall times and parallel efficiencies for the CRWENO5 and WENO5 schemes.

10. The solutions are integrated in time with the fourth-order Runge-Kutta scheme until a time of 3.0 with time-step sizes of 0.01 and 0.002 for the coarse and fine grids, respectively. Figure 21 shows the kinetic energy as a function of the wavenumber for the solutions obtained, including zoomed plots showing the moderate and small length scales. We observe that at both problem sizes, the solution obtained with the CRWENO5 scheme has a spectral resolution comparable to that obtained with the WENO5 scheme on the grid that is 1.25³ times finer. We thus evaluate the computational efficiency of the CRWENO5 scheme by comparing it with that of the WENO5 scheme on a finer grid.

Figure 22(a) shows the wall times (for one Runge-Kutta time-integration stage) for the CRWENO5 scheme on the 256³ points grid and the WENO5 scheme on the 320³ points grid. The subdomain sizes vary from 4³ (5³ for WENO5) points for 262, 144 (64³) processors to 32³ (40³ for WENO5) points on 512 (8⁴) processors. Figure 22(b) shows the efficiencies of the CRWENO5 and WENO5 schemes as a function of the subdomain sizes. One can see from both plots that the CRWENO5 scheme does not scale well at small

subdomain sizes; however, it remains more efficient than the WENO5 scheme. Thus, the CRWENO5 scheme is less expensive than the WENO5 scheme for obtaining comparable solutions, even at the smallest subdomain size of 4^3 points per processor. Figure 22(c) shows the Runge-Kutta stage wall times of the CRWENO5 and WENO5 schemes with constant subdomain sizes of 4^3 and 5^3 points per processor, respectively. The problem sizes vary from 32^3 (40^3 for WENO5) points on 8^3 processors to 256^3 (320^3 for WENO5) points on 64^3 processors. The CRWENO5 scheme is observed to scale well and remains less expensive than the WENO5 scheme as the problem size increases.

5 Conclusions

We present scalable and efficient nonlinear compact schemes in this study by implementing a parallel tridiagonal solver. The solver is based on the substructuring or partitioning approach, but we solve the reduced system iteratively to machine zero accuracy. This is possible because of the diagonal dominance of the reduced system that represents the weak coupling of the finite-difference operators between the interface points in neighboring sub-domains. With the initial guess as the solution of the corresponding diagonal system, machine-zero convergence requires few iterations. The diagonal dominance reduces as the subdomain size shrinks (with increasing processors for the same problem size), and consequently the number of iterations required increases. Thus, the relative cost of the compact schemes increases as the subdomain size decreases. The iterative solution of the reduced system does not use a norm-based exit criterion; instead the number of Jacobi iterations is fixed a priori. Thus, the overall solver avoids any collective communications and involves only point-to-point communications between neighboring processors.

We analyze the performance of the CRWENO5 scheme with manufactured solutions involving the advection of density waves. In one and two spatial dimensions, we observe a critical subdomain size: above this size, the compact schemes are more efficient than the corresponding WENO scheme with comparable accuracy and resolution; below this size, it is less expensive to obtain a solution of comparable accuracy with the WENO scheme on more number of grid points. We demonstrate that this critical subdomain size is insensitive to the global problem size; we thus conclude that the algorithm will have a similar performance on larger problem sizes with correspondingly larger number of processors. The higher efficiency of the compact schemes is amplified by the dimensionality for three-dimensional problems, and they are shown to be more efficient than the WENO scheme for the smallest subdomain sizes considered (4^3 points per processor). Although the number of Jacobi iterations is fixed a priori based on an empirical estimate, the difference in the computational efficiencies of the CRWENO5 and WENO5 is large enough to allow specifying a large (possibly redundant) number of Jacobi iterations. Since fifth-order schemes such as the CRWENO5 and WENO5 require at least three ghost points on each physical and parallel-decomposition boundary for data exchange, a subdomain size of 4^3 points is close to the practical lower limit. This implementation of the CRWENO5 renders it more efficient for the entire range of parallelizations—from coarse-grained to fine-grained simulations.

The parallel implementation of the CRWENO5 scheme is evaluated on two physically relevant flow problems—the isentropic vortex convection and the isotropic turbulence decay. It is verified for these problems that the WENO5 scheme requires a larger number of grid points than does the CRWENO5 scheme to yield comparable solutions. The previous conclusions regarding the scaling and computational efficiencies are demonstrated for these problems on $O(10^5)$ processors. Strong scaling studies show that although the CRWENO5 scheme does not scale as well as WENO5 scheme, it remains more efficient in absolute terms. We also observe through weak scaling results that this implementation remains less expensive as the global domain size and the number of processors increase.

To summarize, our parallel implementation of the nonlinear compact scheme differs from previous attempts by satisfying all these conditions: it does not introduce any parallelization-related approximations; the tridiagonal solver is not significantly more complex than the Thomas algorithm on a single processor; the algorithm requires only point-to-point communications and no collective communications; it does not involve any data rearrangement/transposition across processors; and in three (or higher) dimensions it is more computationally efficient than a standard finite-difference scheme of the same accuracy/resolution for

the entire range of domain decomposition sizes. The computational properties of this implementation are independent of the global problem size, and thus it is expected to scale as well for larger problems on larger number of processors (than what is shown here). These properties are demonstrated through the CRWENO5 scheme; however, this implementation is applicable to other nonlinear compact schemes as well (such as the hybrid compact-WENO schemes) that require solutions to tridiagonal systems at each time-integration step/stage.

Acknowledgments

We acknowledge Dr. Paul Fischer (Argonne National Laboratory) for his help and guidance in the parallel implementation of the tridiagonal solver. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

Bibliography

- [1] *ALCF/Mira Web page*, 2013. <https://www.alcf.anl.gov/mira>.
- [2] *ALCF/Vesta Web page*, 2013. <https://www.alcf.anl.gov/vesta>.
- [3] N. A. ADAMS, *Direct numerical simulation of turbulent compression ramp flow*, Theoretical and Computational Fluid Dynamics, 12 (1998), pp. 109–129.
- [4] N. A. ADAMS AND K. SHARIFF, *A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems*, Journal of Computational Physics, 127 (1996), pp. 27–51.
- [5] S. BALAY, J. BROWN, K. BUSCHELMAN, V. ELJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, B. F. SMITH, AND H. ZHANG, *PETSc Users Manual*, Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.
- [6] S. BALAY, J. BROWN, K. BUSCHELMAN, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, B. F. SMITH, AND H. ZHANG, *PETSc Web page*, 2013. <http://www.mcs.anl.gov/petsc>.
- [7] S. BALAY, W. D. GROPP, L. C. MCINNES, AND B. F. SMITH, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser Press, 1997, pp. 163–202.
- [8] D. S. BALSARA AND C.-W. SHU, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, Journal of Computational Physics, 160 (2000), pp. 405–452.
- [9] I. BERMEJO-MORENO, J. BODART, J. LARSSON, B. M. BARNEY, J. W. NICHOLS, AND S. JONES, *Solving the compressible Navier-Stokes equations on up to 1.97 million cores and 4.1 trillion grid points*, in Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis, SC '13, New York, NY, USA, 2013, ACM, pp. 62:1–62:10.
- [10] R. BORGES, M. CARMONA, B. COSTA, AND W. S. DON, *An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws*, Journal of Computational Physics, 227 (2008), pp. 3191–3211.
- [11] C. CANUTO, *Spectral Methods in Fluid Dynamics*, Computational Physics Series, Springer-Verlag, 1988.
- [12] M. CASTRO, B. COSTA, AND W. S. DON, *High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws*, Journal of Computational Physics, 230 (2011), pp. 1766–1792.
- [13] J. CHAO, A. HASELBACHER, AND S. BALACHANDAR, *A massively parallel multi-block hybrid compact-WENO scheme for compressible flows*, Journal of Computational Physics, 228 (2009), pp. 7473–7491.
- [14] B. COCKBURN AND C.-W. SHU, *Nonlinearly stable compact schemes for shock calculations*, SIAM Journal on Numerical Analysis, 31 (1994), pp. 607–627.
- [15] P. COLELLA AND P. R. WOODWARD, *The piecewise parabolic method (PPM) for gas-dynamical simulations*, Journal of Computational Physics, 54 (1984), pp. 174–201.

- [16] A. W. COOK, W. H. CABOT, P. L. WILLIAMS, B. J. MILLER, B. R. D. SUPINSKI, R. K. YATES, AND M. L. WELCOME, *Tera-scalable algorithms for variable-density elliptic hydrodynamics with spectral accuracy*, in Proceedings of the 2005 ACM/IEEE Conference on Supercomputing, SC '05, Washington, DC, USA, 2005, IEEE Computer Society, pp. 60–60.
- [17] X. DENG AND H. MAEKAWA, *Compact high-order accurate nonlinear schemes*, Journal of Computational Physics, 130 (1997), pp. 77–91.
- [18] X. DENG AND H. ZHANG, *Developing high-order weighted compact nonlinear schemes*, Journal of Computational Physics, 165 (2000), pp. 22–44.
- [19] J. J. DONGARRA AND A. H. SAMEH, *On some parallel banded system solvers*, Parallel Computing, 1 (1984), pp. 223–235.
- [20] W. E AND J.-G. LIU, *Essentially compact schemes for unsteady viscous incompressible flows*, Journal of Computational Physics, 126 (1996), pp. 122–138.
- [21] J. A. EKATERINARIS, *Implicit, high-resolution, compact schemes for gas dynamics and aeroacoustics*, Journal of Computational Physics, 156 (1999), pp. 272–299.
- [22] D. FAUCONNIER AND E. DICK, *Spectral analysis of nonlinear finite difference discretizations*, Journal of Computational and Applied Mathematics, 246 (2013), pp. 113–121. Fifth International Conference on Advanced COmputational Methods in ENgineering (ACOMEN 2011).
- [23] P. F. FISCHER, F. P. PREPARATA, AND J. E. SAVAGE, *Generalized scans and tridiagonal systems*, Theoretical Computer Science, 255 (2001), pp. 423–436.
- [24] L. GAMET, F. DUCROS, F. NICOUD, AND T. POINSOT, *Compact finite difference schemes on non-uniform meshes. application to direct numerical simulations of compressible flows*, International Journal for Numerical Methods in Fluids, 29 (1999), pp. 159–191.
- [25] D. GHOSH, *Compact-reconstruction weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, PhD thesis, University of Maryland, College Park, MD, 2013.
- [26] D. GHOSH AND J. D. BAEDER, *Weighted non-linear compact schemes for the direct numerical simulation of compressible, turbulent flows*, Journal of Scientific Computing, Article in press.
- [27] ———, *Compact reconstruction schemes with weighted ENO limiting for hyperbolic conservation laws*, SIAM Journal on Scientific Computing, 34 (2012), pp. A1678–A1706.
- [28] D. GHOSH, S. MEDIDA, AND J. D. BAEDER, *Compact-reconstruction weighted essentially non-oscillatory schemes for the unsteady navier-stokes equations*, in 42nd AIAA Fluid Dynamics Conference and Exhibit, New Orleans, LA, American Institute of Aeronautics and Astronautics, 2012.
- [29] S. GHOSH, *Direct numerical simulation of the interaction of a laser-induced plasma with isotropic turbulence*, PhD thesis, University of Minnesota, Minneapolis, MN, September 2008.
- [30] S. K. GODUNOV, *A difference scheme for numerical computation of discontinuous solutions of equations of fluid dynamics*, Mathematics Sbornik, 47 (1959), pp. 271–306. (In Russian).
- [31] D. GOTTLIEB AND S. A. ORSZAG, *Numerical Analysis of Spectral Methods: Theory and Applications*, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, 1977.
- [32] S. GOTTLIEB, D. I. KETCHESON, AND C.-W. SHU, *High order strong stability preserving time discretizations*, Journal of Scientific Computing, 38 (2009), pp. 251–289.
- [33] A. HARTEN, *High resolution schemes for hyperbolic conservation laws*, Journal of Computational Physics, 49 (1983), pp. 357–393.

- [34] A. HARTEN, B. ENGQUIST, S. OSHER, AND S. R. CHAKRAVARTHY, *Uniformly high order accurate essentially non-oscillatory schemes, III*, Journal of Computational Physics, 71 (1987), pp. 231–303.
- [35] A. K. HENRICK, T. D. ASLAM, AND J. M. POWERS, *Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points*, Journal of Computational Physics, 207 (2005), pp. 542–567.
- [36] J. HOFHAUS AND E. VAN DE VELDE, *Alternating-direction line-relaxation methods on multicomputers*, SIAM Journal on Scientific Computing, 17 (1996), pp. 454–478.
- [37] G.-S. JIANG AND C.-W. SHU, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1996), pp. 202–228.
- [38] J. W. KIM AND R. D. SANDBERG, *Efficient parallel computing with a compact finite difference scheme*, Computers & Fluids, 58 (2012), pp. 70–87.
- [39] C. B. LANEY, *Computational Gasdynamics*, Cambridge University Press, 1998.
- [40] C. LEE AND Y. SEO, *A new compact spectral scheme for turbulence simulations*, Journal of Computational Physics, 183 (2002), pp. 438–469.
- [41] S. LEE, P. MOIN, AND S. K. LELE, *Interaction of isotropic turbulence with a shock wave*, Tech. Rep. TF-52, Stanford University, Stanford, CA, March 1992.
- [42] S. K. LELE, *Compact finite difference schemes with spectral-like resolution*, Journal of Computational Physics, 103 (1992), pp. 16–42.
- [43] A. LERAT AND C. CORRE, *A residual-based compact scheme for the compressible Navier-Stokes equations*, Journal of Computational Physics, 170 (2001), pp. 642–675.
- [44] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
- [45] X.-D. LIU, S. OSHER, AND T. CHAN, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), pp. 200–212.
- [46] N. N. MANSOUR AND A. A. WRAY, *Decay of isotropic turbulence at low Reynolds number*, Physics of Fluids (1994-present), 6 (1994), pp. 808–814.
- [47] M. P. MARTIN, E. M. TAYLOR, M. WU, AND V. G. WEIRS, *A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence*, Journal of Computational Physics, 220 (2006), pp. 270–289.
- [48] N. MATTOR, T. J. WILLIAMS, AND D. W. HEWETT, *Algorithm for solving tridiagonal matrix problems in parallel*, Parallel Computing, 21 (1995), pp. 1769–1782.
- [49] U. MEIER, *A parallel partition method for solving banded systems of linear equations*, Parallel Computing, 2 (1985), pp. 33–43.
- [50] S. NAGARAJAN, S. K. LELE, AND J. H. FERZIGER, *A robust high-order compact method for large eddy simulation*, Journal of Computational Physics, 191 (2003), pp. 392–419.
- [51] S. PIROZZOLI, *Conservative hybrid compact-WENO schemes for shock-turbulence interaction*, Journal of Computational Physics, 178 (2002), pp. 81–117.
- [52] A. POVITSKY AND P. J. MORRIS, *A higher-order compact method in space and time based on parallel implementation of the Thomas algorithm*, Journal of Computational Physics, 161 (2000), pp. 182–203.
- [53] Y.-X. REN, M. LIU, AND H. ZHANG, *A characteristic-wise hybrid compact-WENO scheme for solving hyperbolic conservation laws*, Journal of Computational Physics, 192 (2003), pp. 365–386.

- [54] P. L. ROE, *Approximate Riemann solvers, parameter vectors, and difference schemes*, Journal of Computational Physics, 43 (1981), pp. 357–372.
- [55] R. S. ROGALLO, *Numerical experiments in homogeneous turbulence*, Tech. Rep. NASA-TM-81315, NASA Ames Research Center, Moffett Field, CA, September 1981.
- [56] T. K. SENGUPTA, G. GANERIWAL, AND S. DE, *Analysis of central and upwind compact schemes*, Journal of Computational Physics, 192 (2003), pp. 677–694.
- [57] J. S. SHANG, *High-order compact-difference schemes for time-dependent Maxwell equations*, Journal of Computational Physics, 153 (1999), pp. 312–333.
- [58] C.-W. SHU, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, Tech. Rep. NASA CR-97-206253 ICASE Report No. 97-65, Institute for Computer Applications in Science and Engineering, November 1997.
- [59] C.-W. SHU, *High order weighted essentially nonoscillatory schemes for convection dominated problems*, SIAM Review, 51 (2009), pp. 82–126.
- [60] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), pp. 439–471.
- [61] ———, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, II*, Journal of Computational Physics, 83 (1989), pp. 32–78.
- [62] R. K. SHUKLA AND X. ZHONG, *Derivation of high-order compact finite difference schemes for non-uniform grid using polynomial interpolation*, Journal of Computational Physics, 204 (2005), pp. 404–429.
- [63] H. S. STONE, *An efficient parallel algorithm for the solution of a tridiagonal linear system of equations*, Journal of the ACM, 20 (1973), pp. 27–38.
- [64] X.-H. SUN AND S. MOITRA, *A fast parallel tridiagonal algorithm for a class of CFD applications*, Tech. Rep. 3585, National Aeronautics and Space Administration, NASA Langley Research Center, Hampton, VA, August 1996.
- [65] B. THORNBURGH, A. MOSEDALE, AND D. DRIKAKIS, *On the implicit large eddy simulations of homogeneous decaying turbulence*, Journal of Computational Physics, 226 (2007), pp. 1902–1929.
- [66] E. F. VAN DE VELDE, *Concurrent Scientific Computing*, Texts in Applied Mathematics, Springer, 1994.
- [67] B. VAN LEER, *Towards the ultimate conservative difference scheme. II: Monotonicity and conservation combined in a second-order scheme*, Journal of Computational Physics, 14 (1974), pp. 361–370.
- [68] M. R. VISBAL AND D. V. GAITONDE, *On the use of higher-order finite-difference schemes on curvilinear and deforming meshes*, Journal of Computational Physics, 181 (2002), pp. 155–185.
- [69] H. H. WANG, *A parallel method for tridiagonal equations*, ACM Transactions on Mathematical Software, 7 (1981), pp. 170–183.
- [70] Z. WANG AND G. P. HUANG, *An essentially nonoscillatory high-order Padé-type (ENO-Padé) scheme*, Journal of Computational Physics, 177 (2002), pp. 37–58.
- [71] R. V. WILSON, A. O. DEMUREN, AND M. CARPENTER, *Higher-order compact schemes for numerical simulation of incompressible flows*, Tech. Rep. NASA/CR-1998-206922, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, February 1998.
- [72] N. K. YAMALEEV AND M. H. CARPENTER, *A systematic methodology for constructing high-order energy stable WENO schemes*, Journal of Computational Physics, 228 (2009), pp. 4248–4272.
- [73] ———, *Third-order energy stable WENO scheme*, Journal of Computational Physics, 228 (2009), pp. 3025–3047.

- [74] H. C. YEE, *Explicit and implicit multidimensional compact high-resolution shock-capturing methods: Formulation*, Journal of Computational Physics, 131 (1997), pp. 216–232.
- [75] S. ZHANG, S. JIANG, AND C.-W. SHU, *Development of nonlinear weighted compact schemes with increasingly higher order accuracy*, Journal of Computational Physics, 227 (2008), pp. 7294–7321.



Mathematics and Computer Science Division

Argonne National Laboratory
9700 South Cass Avenue, Bldg. 240
Argonne, IL 60439

www.anl.gov

Government License: This report has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

Argonne National Laboratory is a U.S. Department of Energy,
Laboratory managed by UChicago Argonne, LLC